Introduction à la Programmation Mathématique

Programmation Linéaire et Optimisation Combinatoire

Y. Hamam & H. Talbot
Groupe E.S.I.E.E. Cité Descartes, BP 99
93162 Noisy-le-Grand CEDEX

7 avril 2009– version 0.2

Table des matières

Introduction

*Cette introduction reprend quasi "in extenso" l'introduction au mémoire HDR (Habilitation à Diriger des Recherches) de Y. Hamam[**?**], d'ailleurs, dans la suite de cet ouvrage nous nous servirons abondemment de ce mémoire sans y faire explicitement référence.*

Les premiers travaux importants sur les techniques d'optimisation datent des années cinquante avec le développement de la programmation mathématique linéaire et non-linéaire. Depuis d'innombrables travaux ont été publiés traitant de problèmes très variés : continus et combinatoires, linéaires et non-linéaires, algébriques ou dans les graphes, déterministes ou stochastiques.

Travailler dans ce domaine nécessite une maîtrise de toute la chaîne : de la modélisation à l'optimisation en passant par l'analyse numérique. Sans un de ces éléments, il est difficile de traiter des problèmes réels. Avant d'entamer la présentation de ces travaux il me semble nécessaire de commencer par quelques remarques d'ordre général pouvant expliquer ma démarche en ce qui concerne la recherche et développement dans ce domaine.

Pour résoudre ce type de problème, le chercheur va être confronté dès l'amorce à une série de choix déterminant pour la suite des opérations. En effet, il lui faut tout d'abord choisir un modèle. Cette opération est des plus délicates dans la mesure ou ce choix va déterminer pour le moins le domaine de validité de la (des) solution(s) obtenue(s). Si le choix se porte sur un modèle complexe (plus précis au sens de la réalité physique) la manipulation risque d'être coûteuse (en temps de calcul) voire hasardeuse. Dans le cas ou un modèle plus simple est utilisé, le risque alors est grand d'obtenir des résultats peu ou pas utilisables. De plus, le choix du modèle peut avoir une influence déterminante sur la pertinence des méthodes de résolutions utilisées. Des exemples de ce type de problème sont d'ailleurs largement abordés dans la suite de ce document.

Lié au choix du modèle une autre considération doit être prise en compte : quid de l'optimalité d'une solution par rapport à un modèle qui ne peut, par nature, représenter totalement la réalité ? Et d'ailleurs, faut il ériger en religion la recherche de La *Solution Optimale* alors que dans beaucoup de cas une *Bonne Solution* est non seulement suffisante mais est même la seule solution raisonnable eu égard aux écarts entre le modèle étudié et le problème réel ? Ne faut il pas plutôt considérer le critère d'optimalité comme un guide vers une *Bonne Solution* ?

Un autre problème doit être évoqué dans cette introduction, problème auquel, d'ailleurs nous n'apportons pas de solution générale, celui de la robustesse d'une solution optimale. En effet, dans les méthodes telles que la programmation linéaire une modification mineure des données peut engendrer une solution optimale radicalement différente de celle obtenue précédemment. Ceci pose de réels problèmes quant à la mise en oeuvre sur des systèmes réels.

Il est aussi essentiel de définir au mieux les critères d'optimalité qui vont permettre de caractériser les solutions. Par exemple, dans le cas des robots manipulateurs deux critères sont couramment utilisés : le temps minimal et l'énergie minimale. En fait, le premier critère va générer des solutions mettant à dure épreuve les capacités mécaniques de la structure alors que le deuxième, qui d'ailleurs ne correspond pas vraiment à l'énergie minimale, va fournir des commandes souples en limitant les efforts sur les actionneurs. Une pondération de ces deux critères fournit des solutions de compromis de bonne qualité. L'optimalité dans cette optique est donc le moyen de permettre à l'utilisateur de choisir à un niveau supérieur, voire stratégique. Nous nous trouvons dans ce cas face à un véritable outil d'aide à la décision.

Dans les premiers temps, les techniques d'optimisations naissantes ont apporté un grand espoir. En effet, les prévisions de puissance croissante des moyens de calcul pouvaient laisser croire qu'en utilisant de telles méthodes la résolution de pratiquement tous les problèmes était envisageable. Cependant, il faut bien constater que cette certitude n'a pas été entièrement réalisée dans les faits et cela pour diverses raisons. Il a été démontré que pour de nombreux problèmes il n'y a pas de possibilité de solutions optimales en temps polynômial ce qui implique l'impossibilité de résolution dans le cas de problèmes de taille réelle. Un des exemples les plus frappants étant le placement des composants et le routage dans le cas des V.L.S.I..

Cette constatation a provoqué le développement de méthodes fournissant des solutions sous-optimales basées sur l'émulation de processus naturels qu'ils soient physiques (recuit simulé) ou biologiques (algorithme génétique ou réseaux neuro-mimétiques). D'une part ces méthodes visent à l'obtention d'une solution présentant un écart limité avec l'optimum avec une probabilité proche de l'unité plutôt que d'assurer la solution optimale. D'autre part elles fournissent un cadre heuristique général permettant l'obtention de solutions ne nécessitant qu'un temps de développement limité.

En tout état de cause et qu'elle que soit la technique retenue les apports croisés entre disciplines et/ou domaines scientifiques prennent ici toute leur importance. La preuve en sera donnée dans la suite de ce mémoire que ce soit au niveau de la confection des modèles ou au niveau de l'utilisation des techniques de résolution. Dans ce type de rapport, on peut d'ailleurs, assez souvent constater un enrichissement mutuel. Ainsi, les réseaux neuro-mimétiques dont l'apport à l'Automatique est aussi important que l'apport de l'Automatique à la compréhension de leur dynamique.

La rédaction de cet ouvrage a tenté de privilégier une approche inductive des diverses méthodes. développées et/ou utilisées par l'auteur. Ainsi nous nous intéressons davantage aux aspects pratiques liés à l'implémentation des méthodes qu'à la démonstration de leurs propriétés. Le lecteur intéressé par une approche plus théorique pourra se reporter avec profit aux divers ouvrages cités dans la bibliographie.

La suite de cet ouvrage est organisée en six chapitres complétés par des annexes.
- Le premier chapitre traite de l'optimisation des problèmes linéaires et ses extensions. Les domaines traités couvrent une gamme d'applications très étendue : réseaux de distribution et de production d'électricité, réseaux de distribution de gaz et d'eau, systèmes de chauffage, robotique,...
- Le deuxième chapitre reprend l'optimisation des problèmes non-linéaires. Les domaines

abordés sont aussi variés que dans le premier chapitre et une attention particulière est portée à l'adéquation entre la formulation du problème et la méthode de résolution adoptée.
- Le troisième chapitre traite de la programmation linéaire en nombres entiers.
- Le quatrième chapitre fait un point sur diverses "méta-heuristiques" : recuit simulé, algorithmes génétiques, réseaux neuro-mimétiques et logique floue.
- Enfin on essaiera de conclure en faisant la synthèse de ces diverses expériences.

Première partie

Théorie, formulation et algorithmes de résolution

Chapitre 1

Modélisation pour la programmation mathématique

Une grande majorité de problèmes d'optimisation dans le milieu industriel se présente en forme d'optimisation combinatoire. Une approximation continue est quelque fois possible, comme dans le cas ou la solution est en nombres entiers mais où les nombres sont suffisamment grands pour permettre une approximation continue. Cependant, pour la plupart des problèmes cette approximation est insuffisante.

Nemhauser et Wolsey[**?**] résument l'importance de la modélisation des problème d'optimisation combinatoire, en général, et des problèmes en nombre entier, en particulier, par la constatation suivante :

"In integer programming, formulating a good model is of crucial importance to solving the model."

Du modèle nous pouvons inférer le type de résolution adéquat. Ce soucis constant est présent tant au niveau de la modélisation mathématique qu'au niveau de sa formulation informatique. En effet, dans le cas de problèmes réels, il serait illusoire de chercher une solution manuellement. L'appel à la programmation est donc un passage obligé.

Exemple 1 : un problème de production

Énoncé

Une entreprise produit trois types de produits (A,B,C) et peut les vendre en quantité illimitée aux prix unitaires suivants :

- A, 10 Francs
- B, 56 Francs
- C, 100 Francs

Les contraintes de productions sont les suivantes :

- Produire une unité de A requiert : une heure de main d'œuvre
- Produire une unité de B requiert : deux heures de main d'œuvre + 2 unités de A
- Produire une unité de C requiert : trois heures de main d'œuvre + 1 unité de B
- Un total de 35 heures de main d'œuvre est disponible

Dans ce premier problème, très simple, nous ne tenons pas compte des coûts induits par la matière première et la main d' œuvre et nous cherchons une solution qui maximise le chiffre d'affaire de l'entreprise en utilisant la main d' œuvre disponible.

Formulation du modèle

Il faut tout d'abord définir nos variables. Soit x_1, x_2 et x_3 les quantités produites de A, B et C respectivement. Il faut noter que :

- la quantité vendue de A est égale à la quantité produite de laquelle on retranche la quantité nécessaire pour produire B soit : $x_1 - 2 \times x_2$
- la quantité vendue de B est égale à la quantité produite de laquelle on retranche la quantité nécessaire pour produire C soit : $x_2 - x_3$

Le chiffre d'affaire de l'entreprise pourra donc s'exprimer de la façon suivante :

$$10 \times (x_1 - 2 \times x_2) + 56 \times (x_2 - x_3) + 100 \times x_3$$

Soit :

$$z = 10x_1 + 36x_2 + 44x_3$$

Nous nous référons à l'expression précédente comme étant la "fonction objectif" ou critère d'optimalité, z étant sa valeur courante. Dans ce cas précis nous souhaitons maximiser la valeur de z et nous parlerons donc d'un problème de maximisation. Nous développons maintenant les contraintes qu'une solution doit respecter.

1. La contrainte sur la main d'œuvre qui doit être inférieure ou égale à 35 heures. Elle s'exprimera de la façon suivante :

$$x_1 + 2x_2 + 3x_3 \leq 35$$

2. Les contraintes sur les relations entre les quantités des différents produits.
   - Il faut deux unités de A pour fabriquer un produit B, donc la quantité fabriquée du produit A doit être supérieure ou égale à deux fois la quantité fabriquée du produit B. Soit

   $$x_1 \geq 2x_2$$

   ou

   $$-x_1 + 2x_2 \leq 0$$

   - de la même façon nous pouvons écrire entre B et C :

   $$x_2 \geq x_3$$

   ou

   $$-x_2 + x_3 \leq 0$$

3. Les contraintes sur les quantités elles-mêmes à savoir qu'il est impensable qu'une quantité fabriquée soit négative. Nous aurons donc :

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0$$

ou bien :

$$x \in \mathbb{R}^{3+}$$

Solution

Dans le cas présent et sans plus de justification on peut dire que la solution optimale à ce problème est la répartition suivante :

$$x_1 = 10$$
$$x_2 = 5$$
$$x_3 = 5$$

On peut remarquer que cette solution est en nombre entiers alors qu'une autre disposition relative aux disposition horaires de la main d'œuvre peut se caractériser par des solutions fractionnaires. En effet, si nous remplaçons 35 par 40 nous obtenons :

$$x_1 = \frac{80}{7}$$
$$x_2 = \frac{40}{7}$$
$$x_3 = \frac{40}{7}$$

Faut-il en conclure que l'optimisation combinatoire milite pour les 35 heures ? De toutes façons résoudre ce problème ne relève pas de la programmation linéaire mais de la programmation linéaire en nombres entiers. Il faut en effet en ce cas remplacer la contrainte $x \in \mathbb{R}^{3+}$ par $x \in \mathbb{Z}^{3+}$

Exemple 2 : répartition de tâches sur des machines

Enoncé

Un ingénieur système est confronté au problème suivant. Soit un ensemble de tâches indépendantes et un ensemble de machines dont les caractéristiques sont :

- pour les tâches :
  - le besoin en capacité de calcul
  - le besoin en mémoire
- pour les machines :
  - la capacité de calcul
  - la capacité mémoire

Un coût sera désigné pour l'affection d'une tâche spécifique à une machine particulière. Il doit écrire un programme répartissant au mieux les tâches sur les machines, optimisant le coût global de l'affectation tout en respectant les contraintes précédentes.

Formulation du modèle

Soit la variable x_{ij} telle que :

$$x_{ij} = \begin{cases} 1 & \text{Si la tâche } i \text{ est affectée à la machine } j \\ 0 & \text{Sinon} \end{cases}$$

Soit T l'ensemble des tâches, et N l'ensemble des machines. Le coût de répartition de l'ensemble des tâches pourra s'exprimer de la façon suivante :

$$z = \sum_{i \in T} \sum_{j \in N} c_{ij} x_{ij}$$

où c_{ij} représente le coût d'affectation de la tâche i sur la machine j.
Les contraintes pourront s'exprimer de la façon suivante :

– Chaque tâche doit être affecté qu'à une et une seule machine. Soit :

$$\sum_{j \in N} x_{ij} = 1 \ \forall i \in T$$

– La capacité mémoire de chaque machine doit être respectée. Soit :

$$\sum_{i \in T} m_i x_{ij} \leq M_j \ \forall j \in N$$

où m_i représente la charge mémoire exigée par la tâche i et M_j la capacité mémoire de la machine j.

– La puissance de calcul de chaque machine doit également être respectée. Soit :

$$\sum_{i \in T} p_i x_{ij} \leq P_j \ \forall j \in N$$

où p_i représente la puissance de calcul exigée par la tâche i et P_j la puissance de calcul disponible sur la machine j.

Envisageons le simple problème de l'affectation de 3 tâches sur deux machines. Il vient :

$$
\begin{array}{lrcrcrcrcrcrcl}
\min z = & x_{11} & + & x_{12} & + & x_{21} & + & x_{22} & + & x_{31} & + & x_{32} & & \\
\text{avec} & & & & & & & & & & & & & \\
 & x_{11} & + & x_{12} & & & & & & & & & = & 1 \\
 & & & & & x_{21} & + & x_{22} & & & & & = & 1 \\
 & & & & & & & & & x_{31} & + & x_{32} & = & 1 \\
 & m_1 x_{11} & & & + & m_2 x_{21} & & & + & m_3 x_{31} & & & \leq & M_1 \\
 & & & m_2 x_{12} & & & + & m_2 x_{22} & & & + & m_3 x_{32} & \leq & M_2 \\
 & p_1 x_{11} & & & + & p_2 x_{21} & & & + & p_3 x_{31} & & & \leq & P_1 \\
 & & & p_1 x_{12} & & & + & p_2 x_{22} & & & + & p_3 x_{32} & \leq & P_2
\end{array}
$$

Dans le premier exemple présenté nous avions un problème qui pouvaient s'exprimer soit en nombres réels soit en nombres entiers. Dans ce second exemple les variables s'expriment en binaires (0-1).

Exemple 3 : production d'énergie électrique

Enoncé

Une compagnie de production d'énergie électrique dispose d'un parc de centrales dont elle veut minimiser les coûts d'exploitation. En première approximation on considérera que le coût de fonctionnement de chacune des unités est proportionnel à la puissance produite. Les contraintes à respecter sont les suivantes :

– la compagnie doit fournir à ses clients une puissance donnée.
– chacune des unités de production à une capacité maximale à respecter.

Formulation du modèle

Soit :

– x_i la puissance délivrée par l'unité i (exprimée en MW)

– P_i la capacité maximale de l'unité i (exprimée en MW)
– c_i la constante de coût de l'unité i (exprimée en F/MWH)
– P_f la puissance à fournir aux clients (exprimée en MW)
– n le nombre d'unité de production

Le coût d'exploitation de l'ensemble des centrales pourra s'exprimer de la façon suivante :

$$z = \sum_i c_i \times x_i$$

Les contraintes, quant à elles, s'exprimerons de la façon suivante :

– la contrainte de production donnera l'équation :

$$\sum_i x_i = P_f$$

– le respect de la capacité maximale de chaque unité de production s'exprimera sous la forme :

$$0 \leq x_i \leq P_i \; \forall i \in [1, n]$$

donc

$$x_i \in \mathbb{R}^{+n}$$

Si on considère le cas de trois unités ayant les caractéristiques suivantes.

Unité	c_i	P_i
1	3	30 MW
2	5	20 MW
3	4	25 MW

On prendra $P_f = 49$MW.
La formulation précédente nous donnera donc :

$$\begin{array}{lcccccccc}
\min z & = & 3x_1 & + & 5x_2 & + & 4x_3 & & \\
\text{avec} & & & & & & & & \\
 & & x_1 & + & x_2 & + & x_3 & = & 49 \\
 & & x_1 & & & & & \leq & 30 \\
 & & & & x_2 & & & \leq & 20 \\
 & & & & & & x_3 & \leq & 25 \\
 & & x_1 & , & x_2 & , & x_3 & \geq & 0
\end{array}$$

Exemple 4 : production d'énergie électrique (suite)

Enoncé

Dans l'exemple précédent nous avons fait deux simplifications importantes :

– Le coût est simplement proportionnel à la puissance fournie par l'unité.
– La puissance fournie par l'unité varie entre 0 et la capacité maximale.

Nous allons considérer maintenant que l'unité ne peut fonctionner qu'entre deux valeurs limites P_{imin} et P_{imax}. Nous considérons également qu'il y a un coût induit dès que l'unité tourne (même si la puissance fournie est nulle). Soit :

$$\text{Coût de l'unité } i = \begin{cases} a_i + c_i x_i & \text{si l'unité } i \text{ tourne} \\ 0 & \text{si l'unité } i \text{ est à l'arrêt} \end{cases}$$

Nous allons donc devoir reformuler le problème précédent.

Formulation du modèle

Soit y_i une variable binaire telle que :

$$y_i = \begin{cases} 1 \text{ si l'unité } i \text{ tourne} \\ 0 \text{ si l'unité } i \text{ est à l'arrêt} \end{cases}$$

Nous pouvons exprimer le coût de fonctionnement de l'unité i de la façon suivante :

$$a_i y_i + c_i x_i$$

avec la condition supplémentaire :

$$P_{imin} y_i \leq x_i \leq P_{imax} y_i$$

Il faut noter que cette formulation conserve la linéarité des équations. Nous pouvons ensuite reformuler notre problème.

Le coût d'exploitation de l'ensemble des centrales pourra s'exprimer de la façon suivante :

$$z = \sum_i (a_i y_i + c_i x_i)$$

Les contraintes, quant à elles, s'exprimerons de la façon suivante :

– la contrainte de production donnera l'équation :

$$\sum_i x_i = P_f$$

– le respect des limites de fonctionnement des unités de production s'exprimera sous la forme :

$$P_{imin} y_i \leq x_i \leq P_{imax} y_i \; \forall i \in [1, n]$$

Si on considère le cas de trois unités ayant les caractéristiques suivantes.

Unité	a_i	c_i	P_{imin}	P_{imax}
1	15	3	10 MW	30 MW
2	10	5	7 MW	20 MW
3	13	4	8 MW	25 MW

Ce qui nous donnera :

$$\begin{array}{lcrcrcrcrcrcrcr}
\min z & = & 3x_1 & + & 5x_2 & + & 4x_3 & + & 15y_1 & + & 10y_2 & + & 13y_3 & & \\
\text{avec} & & & & & & & & & & & & & & \\
 & & x_1 & + & x_2 & + & x_3 & & & & & & & = & 49 \\
 & & x_1 & & & & & - & 30y_1 & & & & & \leq & 0 \\
 & & & & x_2 & & & & & - & 20y_2 & & & \leq & 0 \\
 & & & & & & x_3 & & & & & - & 25y_3 & \leq & 0 \\
 & - & x_1 & & & & & + & 10y_1 & & & & & \leq & 0 \\
 & & & - & x_2 & & & & & + & 7y_2 & & & \leq & 0 \\
 & & & & & - & x_3 & & & & & + & 8y_3 & \leq & 0
\end{array}$$

Remarquons que dans cet exemple nous avons un mélange de variables continues (x_i) et de variables binaires (y_i).

Dans ce chapitre nous avons présenté des problèmes qui se modélisent aisément sous forme algébrique. Ceci étant, les variables appartiennent à des espaces différents puisque nous avons envisagé :

1. des problèmes concernant uniquement des variables continues
2. des problèmes concernant uniquement des variables en nombres entiers.
3. des problèmes concernant uniquement des variables en nombres entiers mais limitées à 0-1 (binaires).
4. des problèmes mélangeant variables continues et variables en nombres entiers (binaires).

Nous verrons par la suite que l'optimisation combinatoire peut également se modéliser en forme de graphe. Nous traiterons cet aspect ultérieurement.

Chapitre 2

Programmation linéaire

Proposée par G.B.Dantzig en 1949 [2] la programmation linéaire a permis la résolution des problèmes d'optimisation dans la plupart des domaines. Utilisée directement ou comme partie d'un autre algorithme, des problèmes de tailles très variées (quelques variables jusqu'à quelques dizaines de milliers) ont été résolus par cette méthode. Pour que ce type de résolution soit efficace, cette méthode nécessite une formulation adéquate tenant compte de la spécificité de chaque problème traité.

La programmation linéaire (PL) a bien évidemment un intérêt en tant que telle, mais encore plus en considérant ses techniques de résolution en tant qu'introduction à celle de problèmes plus contraints, tels la programmation linéaire en nombres entiers.

Il est donc absolument nécessaire de passer par l'apprentissage de la formulation et la résolution des problèmes de PL.

2.1 Un exemple : formulation, représentation et résolution

Quels types de problèmes se prêtent à une formulation et une résolution par programmation linéaire ? Il est utile de commencer par un exemple simple illustratif inspiré par J. L. Casty [1].

2.1.1 Le problème de la nourriture pour chiens

Imaginons la situation suivante :

- Soit une entreprise de nourriture pour chiens, qui fabrique 2 types de granulés : le Wag-Tail (W) et le Bark-Mad (B).
- Chacun des types utilise un mélange de légumes, boeuf et poisson, dans les proportions suivantes :

Ingrédient	Qté totale	Qté dans B	Qté dans W
Légumes	1400 kg	4 kg	4 kg
Poisson	1800 kg	6 kg	3 kg
Bœuf	1800 kg	2 kg	6 kg

- On suppose que la companie opère un bénefice de 12 Euros sur chaque paquet de B et de 8 euros sur ceux de W.
- Comment l'entreprise peut-elle faire pour maximiser son profit ?

Formulation

On note qu'on cherche à maximiser une quantité en fonction de diverses contraintes. Il s'agit bien d'un problème d'optimisation. On peut exprimer le problème de la manière suivante :

- On note par B le nombre de paquet de Bark-Mad produits, et par W le nombre de paquets de Wag-Tail.
- De la table précédente on déduit que la quantité totale de légumes consommés sera de $4W + 4B$ kg, mais qu'on ne peut en aucun cas utiliser plus de 1400 kg de légumes. Donc :

$$4B + 4W \leq 1400 \tag{2.1}$$

- De même, en interprétant les contraintes sur le poisson et le bœuf, on obtient :

$$6B + 3W \leq 1800 \tag{2.2}$$

- et

$$2B + 6W \leq 1800 \tag{2.3}$$

- finalement B et W sont forcément tous deux positifs.
- Du point de vue de l'optimisation du profit, on peut écrire

$$P = 12B + 8W \tag{2.4}$$

On cherche à maximiser P, le profit.

On constate que, sous cette forme, toutes les contraintes ainsi que la fonction de profit s'expriment de manière linéaires. Lorsqu'on cherche à maximiser une telle fonction objectif qui s'exprime de manière linéaire en fonction des variables du problèmes, sous certaines contraintes également exprimée de manière linéaire, on parle de *programmation linéaire*.

Comme il s'agit d'un problème à deux variables (B et W), on peut l'illustrer de manière graphique.

Représentation graphique

Une représentation graphique est en figure 2.1. Dans cette figure, le nombre de paquets de B vendus est en abcisse et le nombre de paquets de W est en ordonnée. On a représenté toutes les contraintes par des droites, en hachurant à chaque fois le demi-plan interdit. C'est ainsi qu'on a 5 contraintes : les contraintes explicites provenant du modèle plus les contraintes de positivité des variables.

Dans cette figure, une solution est un couple (B, W) respectant toutes les contraintes. La zone permise des solutions respectant toutes les contraintes est une intersection de demi-plans, il s'agit donc nécessairement d'une zone convexe. Cette zone est un *simplexe*, c'est-à-dire un polytope[1] convexe de même dimension que l'espace de départ (ici 2), bordée par des hyper-plans de dimension immédiatement inférieure (ici des droites, de dimension 1).

[1] Un polytope est en dimension arbitraire ce qu'un polygone est en dimension 2 et un polyèdre en dimension 3.

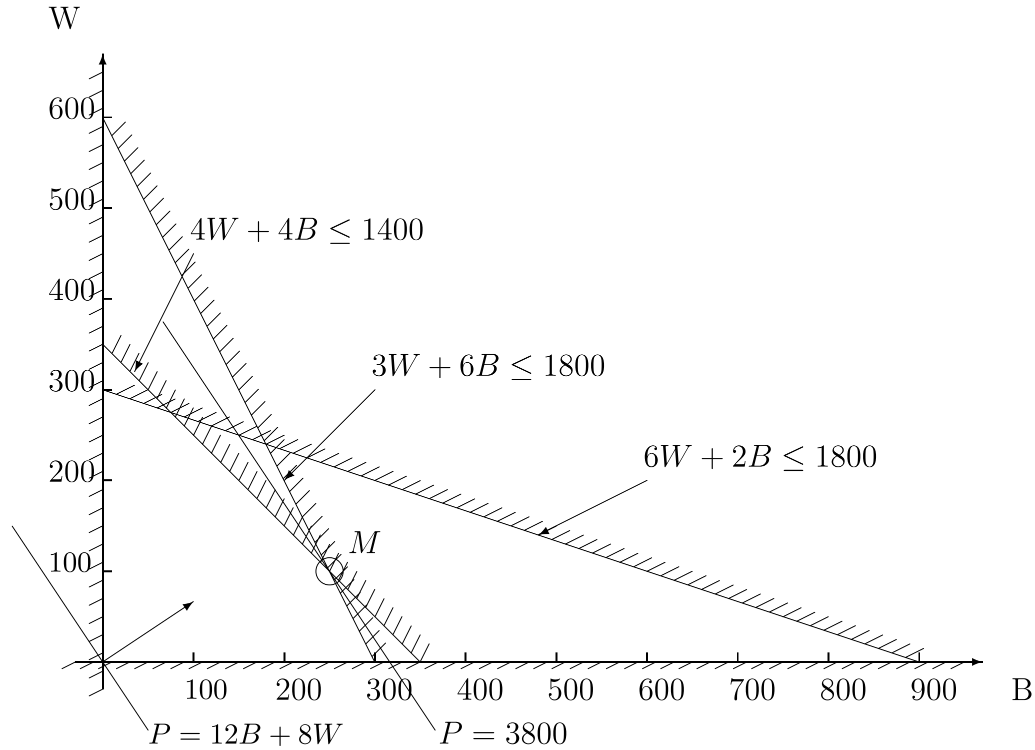


FIG. 2.1 – Exemple de la nourriture pour chiens : l'optimum est en M avec $P = 3800$.

On a d'autre part représenté la fonction de profit (ou *fonction objectif*), que l'on cherche à maximiser, par un ensemble de droites d'équation $12P + 8W = P$, avec P variable. Toutes ces droites sont donc parallèles entre elles. Sur le graphique, on a représenté la droite avec $P = 0$ associée avec un vecteur perpendiculaire à la droite indiquant dans quel sens P grandit.

Optimiser ce système revient donc à trouver P maximal, c'est-à-dire un couple (B, W) tel que P soit le plus grand possible. Un raisonnement géométrique élémentaire nous incite à déplacer la droite $12P + 8W = P$ parallèlement à elle-même, dans le sens où P grandit, jusqu'à trouver l'intersection avec le simplexe telle que P soit le plus grand possible.

Intuitivement, on s'attend, comme c'est le cas ici, à trouver l'optimum sur une frontière du simplexe. On trouve en effet le point M, de coordonnées $(B = 250, W = 100)$, réalisant un profit de $12 * 250 + 8 * 100 = 3800$ Euros, ce qui est l'optimum.

2.1.2 Cas de figure des solutions possibles

On constate que l'intersection extrémale entre la famille des droites de paramètre P et le simplexe est un point. À l'aide de cette représentation graphique, on peut imaginer plusieurs cas de figures :

- Soit une solution optimale existe, et elle est représentée par un point ;
- Soit une famille de solutions existe, dans le cas où une ou plusieurs contraintes sont parallèles à la fonction de profit et bordent le simplexe du côté extrémal : l'ensemble des solution peut alors être représenté par un sous-ensemble de l'ensemble de départ, de dimension $S, 1 \leq S < N$, avec N la dimension de l'espace de départ ;
- Soit une solution existe, et elle est non-bornée, dans le cas où il n'y a pas de contrainte dans la direction où P grandit ;

– Soit toutes les contraintes sont incompatibles, et il n'y a pas de solution.

Dans la suite de cet exemple introductif, on va supposer qu'une solution existe, et qu'elle est réduite à un point.

2.1.3 Recherche d'optimum

Bien entendu, la méthode géométrique pour déterminer l'optimum de ce système fonctionne parfaitement, mais un exemple légèrement plus compliqué, avec 3 produits par exemple, serait difficile à représenter graphiquement : il faudrait avoir recours à la 3ème dimension. Un exemple à 4 dimensions serait extrêmement difficile à résoudre de cette façon. D'autre part, cette méthode ne décrit pas réellement un algorithme.

En revanche, en admettant que la solution optimale est effectivement présente sur la frontière du simplexe, on peut imaginer qu'au lieu d'explorer tout l'espace continu du simplexe, on puisse se contenter de rechercher l'optimum sur ses mêmes frontières. On passe donc d'une recherche dans un espace de dimension N à une recherche dans un espace de dimension $N-1$. De manière récursive, en supposant que dans le cas général aucun des espaces de dimension $N-1$ n'est parallèle aux fonctions paramètrées qu'on cherche à optimiser, on peut se contenter de rechercher l'optimum sur les frontières de ses mêmes sous-espaces, de dimension $N-2$, et ainsi de suite.

De cette manière, on finit par rechercher l'optimum sur un ensemble fini de points (de dimension 0), qui sont les sommets du simplexe. Chacun de ces points forment une *base* du simplexe, et peuvent être trouvé par manipulation du système linéaire d'inéquations qui définit le simplexe. On pourrait imaginer de les énumérer tous et de tester la fonction de profit sur chacun d'entre eux afin de trouver l'optimum. Cependant, le nombre de ces points grandit de manière exponentielle en fonction de la dimensionnalité du problème.

G. Dantzig a le premier proposé de partir d'une base connue du simplexe respectant les contraintes (une *base réalisable*, puis de se déplacer le long des arêtes du simplexe, c'est-à-dire les droites reliant les bases entre elles, de manière à toujours augmenter la fonction de profit le plus rapidement possible, tout en continuant à respecter les contraintes. En procédant de cette manière, de base réalisable en base réalisable, on constate qu'on arrive le plus souvent à trouver l'optimum en un nombre de déplacements proportionnels à la dimensionnalité du problème, et non plus exponentiel en fonction de celle-ci.

2.1.4 Un algorithme élémentaire du simplexe

La méthode que G. Dantzig a développé consiste en une manipulation explicite des inégalités pour trouver les solutions de base réalisables, qu'on appelle maintenant *méthode du simplexe*. Nous la présentons ici en suivant l'exemple de la nourriture pour chiens. Cette méthode, à base de *tableaux*, ne se prête pas très facilement à une mise en œuvre informatique, mais est celle qui est la plus souvent présentée dans les textes classiques sur la programmation linéaire. Il est donc utile de l'avoir vue au moins une fois.

L'idée générale sera reprise au cours des exemples des sections suivantes et au moment du déroulement de l'algorithme que nous proposons. Elle se décompose en les phases suivantes :

Étapes de l'algorithme

1. Partir d'une solution réalisable et calculer la fonction objectif en ce point ;

2. Trouver les arêtes de frontières de l'ensemble réalisable passant par ce point, calculer si la fonction objectif s'améliore en se déplaçant le long de cette arête ;
3. Se déplacer le long de l'arête donnant la plus grande augmentation ;
4. Répéter (2) et (3) jusqu'à ce que la fonction objectif n'augmente plus. On a trouvé la solution optimale.

Variables artificielles

Pour pouvoir manipuler plus facilement les inégalités, on les transforme en égalités par ajout de variables artificielles $\{s_1, s_2, s_3\}$, c'est ainsi que les trois contraintes principales deviennent :

$$\begin{aligned}
4B + 4W + s_1 &= 1400 \\
6B + 3W + s_2 &= 1800 \\
2B + 6W + s_3 &= 1800
\end{aligned}$$

Terminologie

- L'ensemble des valeurs variables $\{B, W, s_i, i \in \{1, \dots 3\}\}$ respectant toutes les contraintes forme une *solution réalisable.*
- Une solution avec m équations et n inconnues, $n \geq m$ avec certaines de ces variables à 0 est une *solution de base*
- L'ensemble des m variables non à zero forme une *base.*

Choix initial

- Pour former une solution de base, on doit choisir 3 variables et mettre les autres à 0.
- On peut poser $B = W = 0$ et résoudre pour les s_i
- Cela donne :

$$\begin{aligned}
s_1 &= 1400 - 4B - 4W \\
s_2 &= 1800 - 6B - 3W \\
s_3 &= 1800 - 2B - 6W \\
P &= 12B + 8W
\end{aligned}$$

- Géométriquement, cette solution est réalisable et correspond à l'origine. Malheureusement pour ce choix, $P = 0$, ce qui n'est clairement pas optimal.

Itérations

De manière itérative, on cherche à améliorer cette solution.

Première itération

- Pour augmenter P, on peut choisir d'augmenter B ou W. Comme B offre le plus grand gain, on choisit cette variable. W reste à 0.

- On introduit donc B dans la base, mais B ne peut pas être plus grand que 300, sinon s_2 deviendrait négative. On choisit donc $B = 300$, ce qui induit $s_2 = 0$.
- On réexprime le système avec ces données :

$$\begin{aligned}
s_1 &= 200 + \frac{2}{3}s_2 - 2W \\
B &= 300 - \frac{1}{6}s_2 - \frac{1}{2}W \\
s_3 &= 1200 + \frac{1}{3}s_2 - 5W \\
P &= 3600 - 2s_2 + 2W
\end{aligned}$$

- Maintenant le profit est de 3600.

Seconde itération

- On peut encore faire mieux en augmentant W, mais pour que s_1 reste positive, W ne peut pas être plus grand que 100.
- On introduit donc W dans la base, avec $W = 100$ ce qui induit $s_1 = 0$.
- On réexprime le système avec ces données :

$$\begin{aligned}
W &= 100 - \frac{1}{2}s_1 + \frac{1}{3}s_2 \\
B &= 250 + \frac{1}{4}s_1 - \frac{1}{3}s_2 \\
s_3 &= 700 + \frac{5}{2}s_1 - \frac{4}{3}s_2 \\
P &= 3800 - s_1 - \frac{4}{3}s_2
\end{aligned}$$

- Maintenant le profit est de 3800. Avec $s_1 = s_2 = 0$, on trouve $(B = 250, W = 100)$ ainsi que $s_3 = 700$.

Optimum

- On ne peut plus augmenter le profit P, car augmenter s_1 ou s_2 diminue P.
- On a donc trouvé l'optimum, qui est le même que celui trouvé graphiquement !
- On peut interpréter le fait que s_1 (légumes) et s_2 (poisson) soient à 0 par le fait que ces ressources sont complètement consommées, ce qui n'est pas le cas de s_3 (boeuf). À l'optimum, il reste encore 700kg de bœuf non consommé.

2.1.5 Résumé

En résume, la méthode du simplexe permet d'optimiser les problèmes de programmation linéaire. La méthode a une interprétation géométrique simple, et donne lieu à un algorithme. En pratique, quelques itérations suffisent pour obtenir l'optimum quand il existe. Nous verrons dans la suite du document l'algorithme en détail, y compris les cas limites.

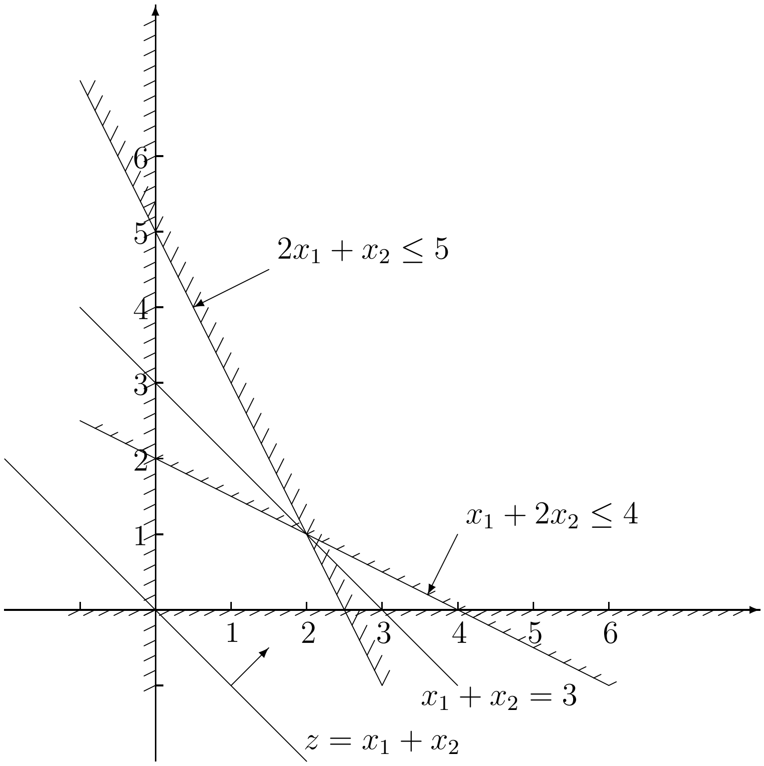


FIG. 2.2 – exemple 2.1

2.2 Cas limites, par l'exemple et approche géométrique

Dans cette section nous allons présenter à travers des exemples quelques cas particuliers qui nous aideront à appréhender certaines propriétés générales des problèmes linéaires. Ceci nous permettra de définir quelques conditions qu'un algorithme de résolution doit satisfaire.

Exemple 2.1

Soit le problème suivant :

$$
\begin{array}{lcrcrcl}
\max z & = & x_1 & + & x_2 & & \\
\text{avec} & & & & & & \\
 & & x_1 & + & 2x_2 & \leq & 4 \\
 & & 2x_1 & + & x_2 & \leq & 5 \\
 & & x_1 & & & \geq & 0 \\
 & & & & x_2 & \geq & 0
\end{array}
$$

En examinant la figure 2.2 nous pouvons aisément remarquer que la solution optimale se trouve au point $x_1 = 2$ $x_2 = 1$ pour un coût de la fonction objectif égal à 3.

A partir de cet exemple nous pouvons noter les caractéristiques suivantes de la solution optimale.

- L'ensemble des solutions est convexe. Nous nous référerons à cet ensemble en l'appelant polytope.
- La solution optimale se trouve sur un point extrême de cet ensemble.
- Pour trouver la solution optimale il suffit d'explorer les points extrêmes du polytope.

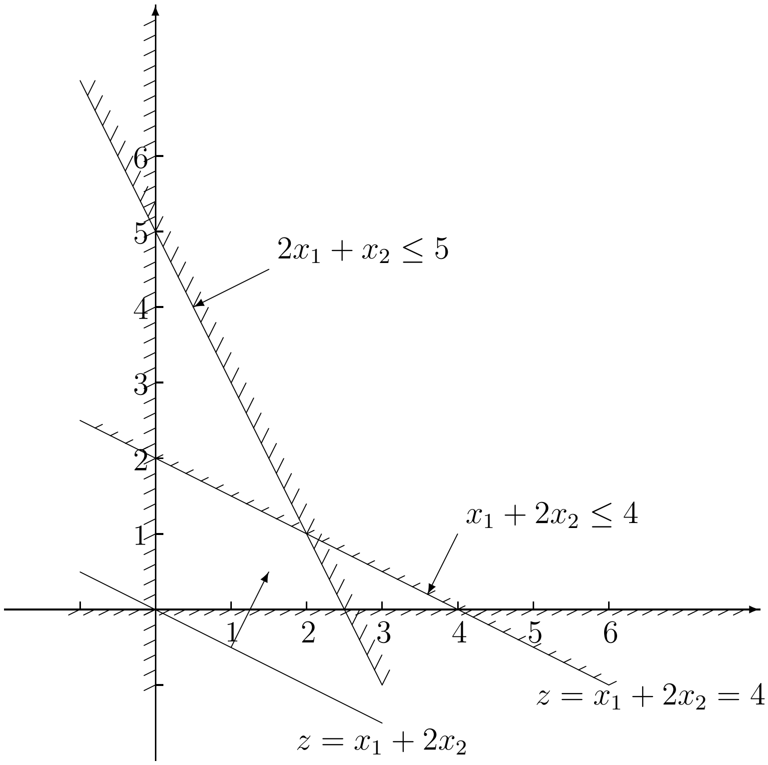


FIG. 2.3 – exemple 2.2

On peut également remarquer que dans ce cas précis la solution optimale est unique.

Exemple 2.2

Soit le problème suivant :

$$
\begin{array}{lcrcrcl}
\max z & = & x_1 & + & 2x_2 & & \\
\text{avec} & & & & & & \\
 & & x_1 & + & 2x_2 & \leq & 4 \\
 & & 2x_1 & + & x_2 & \leq & 5 \\
 & & x_1 & & & \geq & 0 \\
 & & & & x_2 & \geq & 0
\end{array}
$$

Contrairement à l'exemple précédent nous pouvons noter, en nous référant à la figure 2.3 que dans ce cas, la solution optimale est multiple et se trouve sur le segment de droite défini par les points extrêmes suivants : ($x_1 = 0$,$x_2 = 2$) et ($x_1 = 2$, $x_2 = 1$). Cependant nous pouvons choisir un des deux points extrêmes comme solution optimale.

Exemple 2.3

Soit le problème suivant :

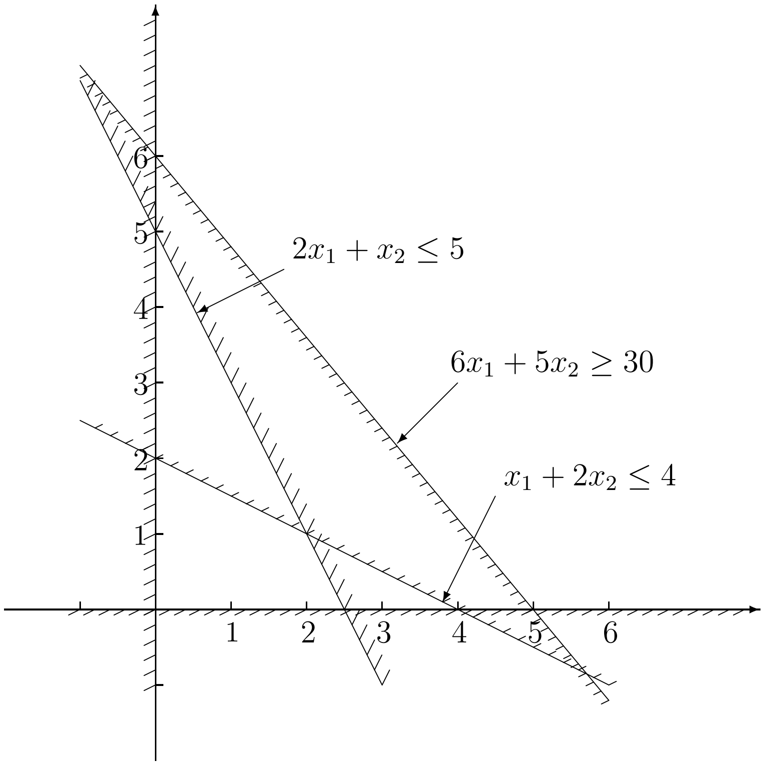


FIG. 2.4 – exemple 2.3

$$
\begin{array}{lrcrcr}
\max z = & x_1 & + & x_2 & & \\
\text{avec} & & & & & \\
& x_1 & + & 2x_2 & \leq & 4 \\
& 2x_1 & + & x_2 & \leq & 5 \\
& 6x_1 & + & 5x_2 & \geq & 30 \\
& x_1 & & & \geq & 0 \\
& & & x_2 & \geq & 0
\end{array}
$$

Dans ce cas précis la représentation graphique, en figure 2.4 indique clairement qu'il n'existe pas de solutions satisfaisant l'ensemble des contraintes.

Exemple 2.4

Soit le problème suivant :

$$
\begin{array}{lrcrcr}
\max z = & x_1 & + & x_2 & & \\
\text{avec} & & & & & \\
& x_1 & + & 2x_2 & \geq & 4 \\
& 2x_1 & + & x_2 & \geq & 5 \\
& x_1 & & & \geq & 0 \\
& & & x_2 & \geq & 0
\end{array}
$$

Enfin, pour ce dernier exemple on voit clairement sur la figure 2.5 que la solution optimale est non bornée puisque tout couple de valeur x_1, x_2 tel que $x_1 + x_2 \geq 5$ est solution.

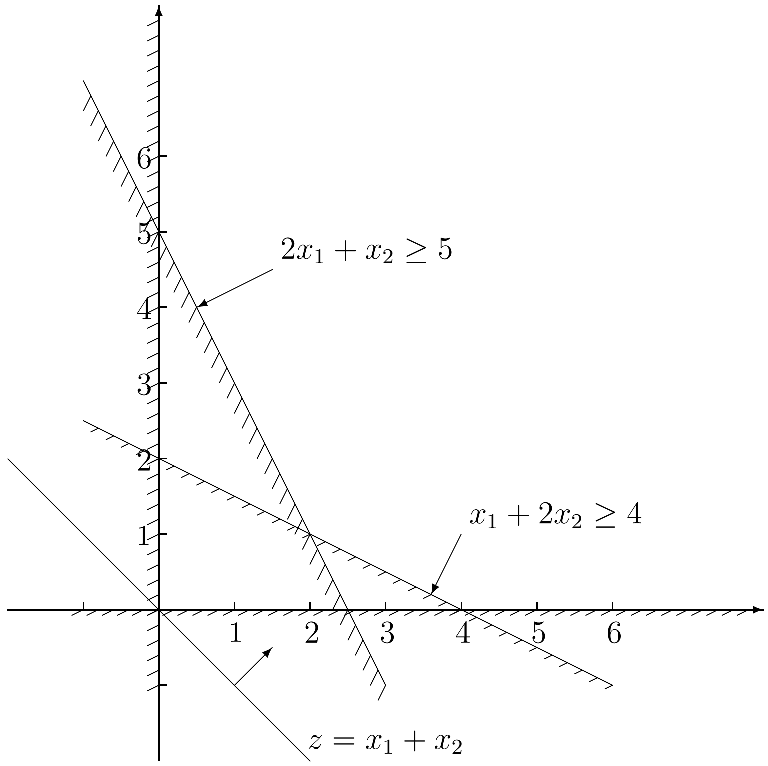


FIG. 2.5 – exemple 2.4

Nous pouvons conclure à partir de ces 4 exemples que tout algorithme prétendant résoudre un problème de programmation linéaire doit pouvoir détecter les situations suivantes :

- Trouver la solution optimale et prouver son unicité.
- Trouver une solution optimale et la multiplicité de l'optimum.
- Détecter l'absence de solutions.
- Détecter le cas où la solution optimale est non bornée.

Au départ nous allons supposer qu'une solution existe. Nous traiterons ensuite les cas où nous n'avons pas de solution de départ. Une extension de la méthode nous permettra ensuite de rechercher une telle solution et d'en détecter l'existence.

La méthode présentée dans la suite est connue sous l'appellation "Algorithme du Simplexe". Avant de présenter l'algorithme nous présenterons un ensemble d'éléments nécessaires à son développement.

2.3 Algorithme du simplexe

Dans cette section, nous allons développer un algorithme du simplexe à partir des principes élémentaires de l'algèbre linéaire. Nous souhaitons déboucher sur un algorithme simple et facile à mettre en œuvre. Pour ce faire, nous devons passer par plusieurs étapes de formalisation des problèmes de programmation linéaire.

2.3.1 Forme standard

La résolution de la programmation linéaire est basée sur la représentation du problème en forme standard. Considérons l'exemple suivant :

Exemple 2.5

Soit le probléme suivant :

$$
\begin{array}{lcrcrcrcl}
\min z & = & 4x_1 & + & 5x_2 & + & x_3 & & \\
\text{avec} & & & & & & & & \\
& & & & 2x_2 & - & x_3 & \geq & 4 \\
& & x_1 & - & 2x_2 & & & \leq & 3 \\
& & x_1 & - & 3x_2 & + & 4x_3 & = & 5 \\
& & x_1 & , & x_2 & , & x_3 & \geq & 0
\end{array}
$$

Forme initiale

Ce problème est dit sous *forme initiale* P_0, qui s'exprime généralement par

Minimiser ou maximiser $f(x), x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, sous les contraintes :

$$
\begin{cases}
g_i & = 0 \quad i \in \{1, \ldots, l_0\} \\
g_j & \leq 0 \quad j \in \{l_0 + 1, \ldots, l_-\} \\
j_k & \geq 0 \quad k \in \{l_-, \ldots, p\}, \text{ avec } \forall i, x_i \geq 0
\end{cases}
$$

avec f, g_i, g_j, g_k des fonctions linéaires des variables x_i.

Forme standard

En introduisant dans l'exemple des variables (positives) x_4 et x_5 dans les premières et deuxièmes inégalités, nous obtenons la forme suivante :

$$
\begin{array}{lcrcrcrcrcrcl}
\min z & = & 4x_1 & + & 5x_2 & + & x_3 & & & & & & \\
\text{avec} & & & & & & & & & & & & \\
& & & & 2x_2 & - & x_3 & - & x_4 & & & = & 4 \\
& & x_1 & - & 2x_2 & & & & & + & x_5 & = & 3 \\
& & x_1 & - & 3x_2 & + & 4x_3 & & & & & = & 5 \\
& & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geq & 0
\end{array}
$$

Dans ce cas nous pouvons utiliser la formulation matricielle suivante qui définit la forme standard P_1 de manière compacte :

$$
\begin{array}{lrcl}
\min & c^t x & & \\
& Ax & = & b \\
& x & \geq & 0
\end{array}
$$

Avec pour cet exemple précis :

$$
A = \begin{bmatrix} 0 & 2 & -1 & -1 & 0 \\ 1 & -2 & 0 & 0 & 1 \\ 1 & 3 & 4 & 0 & 0 \end{bmatrix}, b = \begin{bmatrix} 4 \\ 3 \\ 5 \end{bmatrix}, c^t = \begin{bmatrix} 4 & 5 & 1 & 0 & 0 \end{bmatrix}
$$

Remarquons que la matrice A[2] ainsi obtenue ne présente pas les caractéristiques habituelles des matrices que nous avons coutumes de manipuler dans la mesure où il y a peu de chances qu'elle soit carrée.

[2] de dimensions 3×5 dans notre exemple

Proposition 2.3.1 (Passage à la forme standard)**.** *Tout problème de programmation linéaire sous la forme P_0 peut s'exprimer sous la forme P_1.*

Pour cela, il faut et il suffit :

1. d'ajouter des variables d'excès précédées d'un signe $-$ pour toute contrainte $\geq$;
2. d'ajouter des variables de manque précédées d'un signe $+$ pour les contraintes $\leq$;
3. si la fonction objectif est à maximiser, de la multiplier par -1 et de la changer en une fonction à minimiser.
4. que toutes les variables x_i obéissent à la contrainte $x_i \geq 0$.

les variables d'excès et de manque sont généralement appelées *variables d'écart.*

2.3.2 Solutions de Base

Comme nous l'avons vu en section 2.1.4 la méthode du Simplexe est basée sur l'exploration des solutions de base. Nous commencerons donc par la définition des base est les solutions associées.

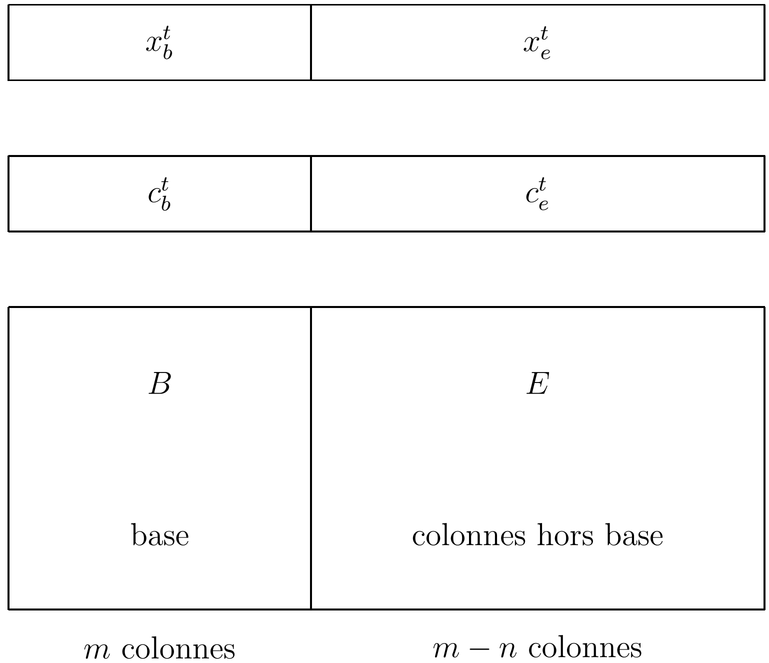
On suppose un problème de programmation linéaire (PL) sous forme standard, comportant n variables $\{x_1, x_2, \ldots, x_n\}$ et m contraintes.

Définition 2.3.2 (Simplexe)**.** *Un polytope convexe, aussi appelé simplexe, est un ensemble* $X = \{x|Ax = b, x \geq 0\}$.

Définition 2.3.3 (Point extrême)**.** *x est un* point extrême *du simplexe X si x ne peut pas être exprimé comme combinaison linéaire d'autres points de X.*

Définition 2.3.4 (Base)**.** *Nous appellerons* base *une sous matrice régulière de A. Il faut donc que la matrice $A(m, n)$ soit de rang m.*

Si nous appelons $\mathcal{B}$ l'ensemble des indices des colonnes de A nous pouvons permuter ces colonnes de façon à obtenir :



Dans cette représentation B est une matrice de dimensions $m \times m$ de colonnes A_k avec $k \in \mathcal{B}$. Nous avons également :

$$A = \begin{bmatrix} B & E \end{bmatrix}, x = \begin{bmatrix} x_b \\ x_e \end{bmatrix}, c^t = \begin{bmatrix} c_b^t & c_e^t \end{bmatrix}$$

Ce qui donne :

$$z = c^t x = c_b^t x_b + c_e^t x_e$$

$$Ax = b$$

$$\begin{bmatrix} B & E \end{bmatrix} \begin{bmatrix} x_b \\ x_e \end{bmatrix} = b$$

$$Bx_b + Ex_e = b$$

Définition 2.3.5 (Solution de base)**.** *Une solution de base est une solution particulière associée à la base B telle que :*

$$\begin{cases} x_e & = & 0 \\ Bx_b & = & b \\ x_b & = & B^{-1}b \end{cases}$$

On dira que x_b est la solution de base associée à la base B

Définition 2.3.6 (Solution de base réalisable)**.** *Une solution de base est dite réalisable (SBR) si l'on a :*

$$x_b = B^{-1}b \geq 0$$

Nous dirons également que la base B associée à cette solution est une base réalisable.

Définition 2.3.7 (Solution de base dégénérée)**.** *Si le vecteur $\bar{b} = B^{-1}b$ contient des termes nuls, on dira que cette solution est une solution de base dégénérée.*

Exemple 2.6

Illustrons ce qui vient d'être dit par le problème suivant :

$$\begin{array}{lcrcrcl} \min & - & x_1 & - & 2x_2 & & \\ \text{avec} & & & & & & \\ & & x_1 & + & 2x_2 & \leq & 4 \\ & & 2x_1 & + & x_2 & \leq & 5 \\ & & x_1 & , & x_2 & \geq & 0 \end{array}$$

La forme standard associée à ce problème est :

$$\begin{array}{lcrcrcrcrcl} \min & - & x_1 & - & 2x_2 & & & & & & \\ \text{avec} & & & & & & & & & & \\ & & x_1 & + & 2x_2 & + & x_3 & & & = & 4 \\ & & 2x_1 & + & x_2 & & & + & x_4 & = & 5 \\ & & x_1 & , & x_2 & , & x_3 & , & x_4 & \geq & 0 \end{array}$$

On peut essayer de constituer une base en utilisant l'ensemble $\mathcal{B} = \{1, 3\}$ ce qui nous donne :

$$B = \begin{bmatrix} A_1 & A_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$$

$$E = \begin{bmatrix} A_2 & A_4 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$$

$$x_b = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}, x_e = \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}$$

L'inverse de B existe, donc B correspond à une base.

$$B^{-1} = \begin{bmatrix} 0 & 1/2 \\ 1 & -1/2 \end{bmatrix}$$

La solution de base correspondante est donc :

$$x_b = B^{-1}b = \begin{bmatrix} 0 & 1/2 \\ 1 & -1/2 \end{bmatrix} = \begin{bmatrix} 5/2 \\ 3/2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} > 0$$

Cette solution est bien une solution de base réalisable.

2.3.3 Caractéristiques d'une solution de base réalisable

On a le théorème important suivant

Théorème 2.3.8 (Caractéristiques d'une solution de base réalisable)**.**
Soit x une solution de bas d'un simplexe X, alors x est réalisable si et seulement si x est un point extrême.

Démonstration. (Partielle)

Soit x une solution de base réalisable. x sera de la forme :

$$x = [x_1, x_2, \ldots, x_m, 0, 0, \ldots, 0]^t$$

Si x n'est pas un point extrême, il existe deux points solutions α et β différents de x et un coefficient λ tels que :

$$x = \lambda\alpha + (1-\lambda)\beta \text{ avec } 0 < \lambda < 1$$

Soit :

$$\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_m, \alpha_{m+1}, \ldots, \alpha_n]^t = \begin{bmatrix} \alpha_b \\ \alpha_e \end{bmatrix}$$

$$\beta = [\beta_1, \beta_2, \ldots, \beta_m, \beta_{m+1}, \ldots, \beta_n]^t = \begin{bmatrix} \beta_b \\ \beta_e \end{bmatrix}$$

Nous avons alors :

$$\lambda\alpha_i + (1-\lambda)\beta_i = 0, \forall i \in [m+1, n]$$

Etant donné qu'on a $\lambda > 0, 1-\lambda > 0, \alpha_i \geq 0$ et $\beta_i \geq 0$ cette expression ne peut-être satisfaite que pour : $\alpha_i = \beta_i = 0$ ce qui donne $\alpha_e = \beta_e = 0$. En remplaçant dans l'équation $Ax = b$ nous avons $\alpha_b = \beta_b = B^{-1}b = x_b$ soit $x = \alpha = \beta$ donc il n'existe pas α et β différent de x donc x est un point extrême. □

Nous venons donc de démontrer que si x est solution de base réalisable, x est un point extrême. pour terminer la démonstration il faut encore démontrer à contrario que si x est un point extrême, il correspond à une solution de base réalisable. Le lecteur intéressé pourra se reporter à Minoux [3] page 35.

Nombre de bases candidates

Le nombre de bases candidates est égale à la combinaison de n variables parmi m, soit $C_n^m = \frac{n!}{(n-m)!m!}$.

Pour l'exemple précédent $\frac{4!}{2!2!} = 6$. Cependant toutes les bases candidats ne sont pas inversibles donc ne peuvent pas être utilisées comme base. Donc on peut dire que le nombre de base est bornée supérieurement par C_n^m. De plus, toutes les bases ne donnent pas des solutions de base réalisables. Dans l'exemple précédent, le nombre de base réalisable (facilement vérifiable) est de 4 et non de 6. Nous pouvons donc conclure qu'une méthode basée sur l'exploration des points extrêmes est *a-priori* une méthode non-polynômiale. La méthode du simplexe que nous allons développer par la suite est effectivement non-polynômiale.

2.3.4 Solution Optimale

Nous avons démontré graphiquement plus haut que la solution optimale dans le cas de deux variables est en un point extrême. Nous allons faire de même pour le cas général.

Théorème 2.3.9. *(Optimum en un point extrême)*
L'optimum de z est atteint en au moins un point extrême du simplexe X.

Démonstration.
Soit $y^1, \ldots, y^T$ l'ensemble des points extrêmes et

$$z^* = min_{i \in [1,T]} \left\{ c^t y^i \right\}$$

Tout autre point x à l'intérieur de l'ensemble des solutions peut être exprimé sous la forme :

$$x = \sum_{i=1}^{T} \lambda_i y^i \text{ et } \sum_{i=1}^{T} \lambda_i = 1 \text{ pour } \lambda_i \geq 0$$

Si nous posons :

$$z(x) = c^t x$$

Il vient :

$$z(x) = c^t \sum_{i=1}^{T} \lambda_i y^i = \sum_{i=1}^{T} \lambda_i z(y^i)$$

Puisque $z(y^i) \geq z^*$ nous aurons :

$$z(x) \geq \sum_{i=1}^{T} \lambda_i z^* = z^* \sum_{i=1}^{T} \lambda_i = z^*$$

Donc tout point intérieur a un coût supérieur ou égal à z^* donc l'optimum se trouve bien en un point extrême. □

Nous allons maintenant développer les conditions qui nous permettrons de caractériser une solution optimale. Ceci est obtenu par ce que nous appelons *les coûts réduits*.

2.3.5 Coûts réduits

Pour une solution de base réalisable nous pouvons écrire :

$$z = c_b^t x_b + c_e^t x_e$$

et

$$B x_b + E x_e = b$$

donc

$$x_b = B^{-1}(b - E x_e)$$

Par simple substitution, nous obtenons,

$$z = c_b^t B^{-1} b + (c_e^t - c_b^t B^{-1} E) x_e$$

Nous poserons :

$$\bar{c}_e^t = (c_e^t - c_b^t B^{-1} E)$$

Donc nous aurons :

$$z = c_b^t B^{-1} b + \bar{c}_e^t x_e$$

Pour cette solution de base réalisable, $x_e = 0$ et z est donc égale à $c_b^t B^{-1} b$. Le deuxième terme correspond à l'augmentation du coût pour une augmentation des variables dans x_e. Chaque éléments de $\bar{c}_e^t$ est égale à l'augmentation du coût pour une augmentation unitaire de la variable correspondante.

C'est pour cette raison que ces éléments sont appelés coûts réduits. Il faut noter que si tout ces coûts sont positifs toute augmentation des variables de x_e augmentera la valeur de z donc la solution obtenue est un optimum local (global vu la convexité de l'ensemble des solutions).

Exemple 2.7

Illustrons avec le problème suivant :

$$\begin{array}{lcccccccccc}
\text{min} & - & x_1 & - & x_2 & & & & & & \\
\text{avec} & & & & & & & & & & \\
& & x_1 & + & 2x_2 & + & x_3 & & & = & 4 \\
& & 2x_1 & + & x_2 & & & + & x_4 & = & 5 \\
& & x_1 & , & x_2 & , & x_3 & , & x_4 & \geq & 0
\end{array}$$

On peut essayer de constituer une base en utilisant l'ensemble $\mathcal{B} = \{1, 2\}$ ce qui nous donne :

$$B = \begin{bmatrix} A_1 & A_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

$$E = \begin{bmatrix} A_3 & A_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$x_b = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, x_e = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

L'inverse de B existe, donc B correspond à une base.

$$B^{-1} = \frac{1}{3} \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix}$$

La solution de base correspondante est donc :

$$x_b = B^{-1} b = \frac{1}{3} \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} > 0$$

Calculons les coûts réduits. Nous avons :

$$c_e^t = \begin{bmatrix} c_3 & c_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

et

$$c_b^t = \begin{bmatrix} c_1 & c_2 \end{bmatrix} = \begin{bmatrix} -1 & -1 \end{bmatrix}$$

donc

$$\bar{c}_e^t = (c_e^t - c_b^t B^{-1} E) = -\frac{1}{3}\begin{bmatrix}-1 & -1\end{bmatrix}\begin{bmatrix}-1 & 2\\ 2 & -1\end{bmatrix} = -\frac{1}{3}\begin{bmatrix}-1 & -1\end{bmatrix} = \begin{bmatrix}\frac{1}{3} & \frac{1}{3}\end{bmatrix}$$

Nous avons donc une base optimale puisque toute augmentation des variables x_3 et x_4 se matérialiserait par une augmentation du coût z (on rappelle qu'on cherche à minimiser z).

2.3.6 Amélioration d'une solution de base

Considérons maintenant le cas où notre base soit telle que $\bar{c}_e^t$ ne soit pas strictement supérieure ou égale à 0. Dans ce cas, il existe au moins une variable x_k de x_e telle que $\bar{c}_k < 0$. Une augmentation de la valeur de x_k est donc de nature à réduire la valeur de z.

Changement de base Pour changer la base et se déplacer le long d'un arête du polytope X, il faut faire entrer une nouvelle variable dans la base, et en faire sortir une autre.

Variable à faire entrer dans la base On choisit de faire entrer dans la base celle des variables hors-base dont le coût réduit est le plus négatif. C'est celle qui donne la meilleure variation, donc celle qui a localement le plus de chance de faire varier notre solution dans le bon sens.

On choisit donc la variable x_k a faire entrer dans la base telle que

$$k = \underset{j}{\operatorname{argmin}}\{\bar{c}_j < 0\}$$

Variable à faire sortir de la base Nous allons donc maintenant partir de la SBR courante le long de l'arête correspondant à l'augmentation de x_k. Géométriquement, la variable sortant de la base est la première qui tombera à zéro lors de ce déplacement.

Si pour quelque variable x_k de x_e, $\bar{c}_k < 0$, la solution peut être améliorée en augmentant x_k

$$x_b = B^{-1}(b - A_k x_k - E' {x'}_e)$$

Ici E' est la matrice des colonnes de A correspondant aux variables hors-base à laquelle on a retiré x_k en sus. De même, ${x'}_e$ est le vecteur de toutes les variables hors-base courantes, avec x_k en moins.

En fixant les ${x'}_e$ à zero et variant x_k seulement

$$x_b = B^{-1}(b - A_k x_k) = B^{-1}b - B^{-1}A_k x_k$$

soit :

$$x_b = \bar{b} - P x_k$$

avec $P = B^{-1}A_k$

$$\begin{bmatrix} x_{b1}\\ x_{b2}\\ .\\ .\\ .\\ x_{bm}\end{bmatrix} = \begin{bmatrix} \bar{b}_1\\ \bar{b}_2\\ .\\ .\\ .\\ \bar{b}_m\end{bmatrix} + \begin{bmatrix} P_1\\ P_2\\ .\\ .\\ .\\ P_m\end{bmatrix} x_k$$

Etant donné que $x_k = 0$, nous pouvons seulement l'augmenter. Nous pouvons considérer deux cas

- **cas 1 :** $\forall i, P_i \leq 0$
  Dans ce cas la solution est non bornée : en faisant tendre x_k vers l'infini, z tend vers moins l'infini.
- **cas 2 :**
  Dans ce cas nous avons de nouveau deux possibilités :
  1. **cas 1 :** pour tout les i tels que $P_i \leq 0$
     Dans ce cas, nous avons $x_{bi} \geq 0$ pour tout $x_k \geq 0$. Ce cas est dit donc non critique car x_{bi} ne peut pas s'annuler, donc x_{bi} ne peut pas sortir de la base.
  2. **cas 2 :** considérons maintenant tous les i tel que $P_i > 0$
     $x_{bi} = \bar{b}_i - P_i x_k$ ce qui donne $x_{bi} \leq 0$ pour $x_k \geq \bar{b}_i / P_i$. Donc pour tout $P_i > 0$, il existe une valeur maximale de x_k $(= \bar{b}_i / P_i)$ permettant $x_b \geq 0$.

On doit faire sortir de la base celle des variables qui s'annule le plus rapidement parmi celles dont le P_i est positif, c'est à dire avec x_k le plus faible possible. Dans le cas général, toutes les autres variables sont inacceptables car elles conduisent à une solution de base *non réalisable* (en dehors du simplexe).

On doit prendre x_j avec j telle que :

$$j = \underset{i/P_i>0}{\operatorname{argmin}} \left[\frac{\bar{b}_i}{P_i} \right].$$

La valeur de x_k vaudra alors

$$x_k = \left[\frac{\bar{b}_j}{P_j} \right]$$

La variable en position j dans la base est alors nulle et remplacée par x_k.

2.3.7 Algorithme du simplexe

Nous sommes maintenant en mesure d'énoncer l'algorithme complet du simplexe, en figure 2.6, donné pour une *minimisation.*

2.3.8 Illustration de l'algorithme

Nous allons illustrer l'algorithme sur un exemple, soit le problème suivant :

Enoncé

- Une ébénisterie produit des bureaux, des tables et des chaises
- Chaque type de produit réclame du bois et deux types de travaux : mise en forme et finition, suivant le tableau :

Ressource	bureau	table	chaise
planches	8m	6m	1m
mise en forme	4h	2h	1.5h
finition	2h	1.5h	0.5h

- On dispose de 48m de planches, 20h de mise en forme et 8h de finition.
- On vend un bureau pour 60 Euros, une table pour 30 Euros et une chaise pour 20 Euros.

Initialisation Soit un problème de programmation linéaire en forme standard

$$\begin{array}{rrcl} \min & c^t x & & \\ & Ax & = & b \\ & x & \geq & 0 \end{array}$$

Soit une solution de base réalisable (SBR) pour ce système avec la base $B = B^0$. On suppose B^{-1} existe et $B^{-1}b \geq 0$. On pose $A = BE$.

Tant que la solution est non-optimale et bornée faire :

1. *Calculer*
   –
   $$\bar{b} = B^{-1}b \text{ (Solution de base réalisable)}$$
   –
   $$\bar{c}_e^T = c_e^T - c_b^T B^{-1} E \text{ (Coûts réduits)}$$
2. – *Si* $\bar{c}_e^T \geq 0$: solution optimale La solution est celle de l'équation **??** ci-dessus.
   – *Sinon* choisir x_l tel que $\bar{c}_l < 0$ la plus grande en valeur absolue (la plus négative).
3. *Calculer*
   $$P = B^{-1}A_l \text{ (colonne l de A)}$$
4. – *Si* $\forall i, P_i \leq 0$: solution non bornée
   – *Sinon :*
   $$x_l = \min_{k/P_k>0} \left\{ \frac{\bar{b}_k}{P_k} \right\} \tag{2.5}$$
   $$j = \operatorname{argmin}_{k/P_k>0} \left\{ \frac{\bar{b}_k}{P_k} \right\} \tag{2.6}$$
   – Remplacer la variable en position j dans la base par x_l.

fin tant que

FIG. 2.6 – Algorithme du simplexe pour une minimisation

– La demande pour les chaises et les bureaux est illimitée, mais on ne pense vendre que 5 tables au plus.

On cherche à maximiser le profit de l'ébénisterie.

Formulation La formulation sous forme P_0 est assez simple, il suffit de lire le tableau ci-dessus par colonnes pour les contraintes, et de multiplier les variables par leur valeur unitaire pour obtenir la fonction-objectif.

– Variables : x_1 = nb. bureaux, x_2 = tables, x_3= chaises
– Objectif : $\max z = 60x_1 + 30x_2 + 20x_3$
– Contraintes :

$$\begin{array}{rcrcrcr} 8x_1 & + & 6x_2 & + & x_3 & \leq & 48 \\ 4x_1 & + & 2x_2 & + & 1.5x_3 & \leq & 20 \\ 2x_1 & + & 1.5x_2 & + & 0.5x_3 & \leq & 8 \\ & & x_2 & & & \leq & 5 \end{array}$$

$x_1, x_2, x_3 \geq 0$

Pour transformer cette formulation en forme standard P_1, on doit introduire 4 variables d'écart, toutes précédée du signe (+), et transformer la maximisation en minimisation. On obtient :

$$\begin{array}{llrcrcrcrcrcrcrr} \min & z = & -60x_1 & - & 30x_2 & - & 20x_3 & & & & & & & & & \\ & & 8x_1 & + & 6x_2 & + & x_3 & + & x_4 & & & & & & & = 48 \\ & & 4x_1 & + & 2x_2 & + & 1.5x_3 & & & + & x_5 & & & & & = 20 \\ & & 2x_1 & + & 1.5x_2 & + & 0.5x_3 & & & & & + & x_6 & & & = 8 \\ & & & & x_2 & & & & & & & & & + & x_7 & = 5 \end{array}$$

Base initiale réalisable Les données s'expriment sous forme matricielle de la manière suivante :

$$A = \begin{bmatrix} 8 & 6 & 1 & 1 & 0 & 0 & 0 \\ 4 & 2 & 1.5 & 0 & 1 & 0 & 0 \\ 2 & 1.5 & 0.5 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 48 \\ 20 \\ 8 \\ 5 \end{bmatrix}$$

$$C^T = \begin{bmatrix} -60 & -30 & -20 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Comme base initiale, on va choisir la base triviale des variables d'écart. C'est une pratique courante, car la matrice correspondante est toujours dérivée de la matrice identité, car les coefficients des variables d'écart sont toujours 1 ou -1. Vérifier qu'il s'agit bien d'une base réalisable est donc simple et rapide.

$$\mathrm{VB}_0 = \{x_4, x_5, x_6, x_7\}$$

On a donc le système suivant :

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 48 \\ 20 \\ 8 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 8 & 6.0 & 1.0 \\ 4 & 2.0 & 1.5 \\ 2 & 1.5 & 0.5 \\ 0 & 1.0 & 0.0 \end{bmatrix}$$

Première itération La première itération de l'algorithme est relativement simple, comme $B^{-1} = B$.

- Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

  $$\begin{array}{cc} & \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ C_e^T = & [\begin{array}{ccc} \text{-60} & \text{-30} & \text{-20} \end{array}] \end{array}$$

  Donc x_1 (variable avec le coefficient le plus négatif) entre dans la base.
- $P = B^{-1}A_1 = [\ 8 \quad 4 \quad 2 \quad 0\ ]^T$
- ratios :

  $$\begin{array}{ccccc} & x_4 & x_5 & x_6 & x_7 \\ \frac{\bar{b}}{P} = & 6 & 5 & 4 & \infty \end{array}$$

  La variable avec le plus petit coefficient positif sort de la base : x_6

Deuxième itération On présente les variables de base en gardant autant que possible l'ordre initial, mais il s'agit bien sûr d'un choix. L'essentiel étant de ne pas se tromper !

- On a maintenant $\text{VB}_1 = \{x_4, x_5, x_1, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 0 & 8 & 0 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 16 \\ 4 \\ 4 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 6 & 1 & 0 \\ 2 & 1.5 & 0 \\ 1.5 & 0.5 & 1 \\ 1 & 0.0 & 0 \end{bmatrix}$$

- Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

  $$\begin{array}{cc} & \begin{array}{ccc} x_2 & x_3 & x_6 \end{array} \\ C_e^T = & [\begin{array}{ccc} 15 & \text{-5} & 30 \end{array}] \end{array}$$

  Donc x_3 (le min) entre dans la base.
- $P = B^{-1}A_3 = [\ -1 \quad 0.5 \quad 0.25 \quad 0\ ]^T$
- ratios :

  $$\begin{array}{ccccc} & x_4 & x_5 & x_1 & x_7 \\ \frac{\bar{b}}{P} = & \text{-16} & 8 & 16 & \infty \end{array}$$

  donc x_5 (le min dont P est positif) sort de la base.

Troisième itération

- On a maintenant $\text{VB}_2 = \{x_4, x_3, x_1, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 1.0 & 8 & 0 \\ 0 & 1.5 & 4 & 0 \\ 0 & 0.5 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 24 \\ 8 \\ 2 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 6 & 1 & 0 \\ 2 & 1 & 0 \\ 1.5 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

- Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

  $$\begin{array}{cc} & \begin{array}{ccc} x_2 & x_5 & x_6 \end{array} \\ C_e^T = & [\begin{array}{ccc} 5 & 10 & 10 \end{array}] \end{array}$$

  Tous les coûts réduits sont positifs, on a trouvé l'optimum !

Solution

- La solution est constituée des variable de base réalisable. Ici

$$\text{VB}_O = \{x_4, x_3, x_1, x_7\}$$

- Les valeurs de ces variables sont données par

$$\bar{b} = \{24, 8, 2, 5\}$$

  respectivement. Toutes les autres valeurs sont à 0.
- La fonction objectif vaut donc :

$$z = -60x_1 - 30x_2 - 20x_3 = -280$$

Discussion On a résolut ici un problème de PL en 4 dimension en 3 itérations.

Chapitre 3

Le simplexe en pratique

Dans ce chapitre nous allons explorer plus en détail comment l'algorithme du simplexe fonctionne. L'algorithme donné au chapitre précédent est correct mais il est nécessaire d'avoir vu les cas limites dans un contexte pédagogique avant de les rencontrer « dans la nature »dans le cadre des applications.

D'autre part, les exemples que nous avons donnés pour le moment permettent d'initialiser facilement l'algorithme. Nous allons voir qu'il n'est pas toujours évident de postuler et d'exhiber une base réalisable initiale. En effet il est facile de donner un simplexe qui ne contient pas l'origine. Dans ce cas la recherche d'une base initiale n'est pas nécessairement simple et peut même ne pas exister.

Nous explorons ensuite les formulations duales/primales. Savoir passer de l'une à l'autre permet en particulier d'exhiber des meilleurs algorithmes et d'encadrer l'optimum efficacement. Un algorithme primal-dual est également moins sujet aux cas difficiles du simplexe.

Enfin nous donnons un exemple de problèmes pour lesquels l'algorithme du simplexe ne donne pas la solution en temps polynomial.

3.1 Cas limites

Ici nous reprenons sous forme algébrique ce que nous avions vu sous forme géométrique au chapitre 2, à savoir les divers cas du simplexe.

3.1.1 Solution unique

Le cas à une seule solution correspond au cas où il existe au moins une solution de base réalisable, et seulement une d'entre elles optimise la solution.

Reprenons l'exemple de l'ébénisterie idéalisée déjà résolu au chapitre précédent :

- Une ébénisterie produit des bureaux, des tables et des chaises
- Chaque type de produit réclame du bois et deux types de travaux : mise en forme et finition, suivant le tableau :

Ressource	bureau	table	chaise
planches	8m	6m	1m
mise en forme	4h	2h	1.5h
finition	2h	1.5h	0.5h

- On dispose de 48m de planches, 20h de mise en forme et 8h de finition.
- On vend un bureau pour 60 Euros, une table pour 30 Euros et une chaise pour 20 Euros.

– La demande pour les chaises et les bureaux est illimitée, mais on ne pense vendre que 5 tables au plus.
– On veut maximiser le profit.

Ici nous allons directement à la 3eitération :

3ème itération On a comme variables de base VB=$\{x_4, x_3, x_1, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 1.0 & 8 & 0 \\ 0 & 1.5 & 4 & 0 \\ 0 & 0.5 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 24 \\ 8 \\ 2 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 6 & 1 & 0 \\ 2 & 1 & 0 \\ 1.5 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cc} & \begin{array}{ccc} x_2 & x_5 & x_6 \end{array} \\ C_e^T = & \begin{bmatrix} 5 & 10 & 10 \end{bmatrix} \end{array}$$

Tous les coûts réduits sont positifs, donc aucun échange de variable ne peut réduire le coût, donc on a donc trouvé l'optimum !

Solution La solution est constituée des variable de base réalisable. Ici

$$\text{VB}=\{x_4, x_3, x_1, x_7\}$$

Les valeurs de ces variables sont données par

$$\bar{b} = \{24, 8, 2, 5\}$$

respectivement. Toutes les autres valeurs sont à 0.

La fonction de coût vaut donc :

$$z = -60x_1 - 30x_2 - 20x_3 = -280$$

Dans le problème initial, le profit est de 280 euros.

On interprète la solution de la façon suivante :

– $x_1 = 2$, on a donc fabriqué 2 bureaux ;
– $x_2 = 0$, on a donc fabriqué 0 table ;
– $x_3 = 8$, on a donc fabriqué 8 chaises ;
– $x_4 = 24$, il reste 24m de planches ;
– $x_5 = 0$, on a consommé toutes les heures de mise en forme ;
– $x_6 = 0$, on a aussi consommé toutes les heures de finition ;
– $x_7 = 5$, on a vendu aucune table, cette contrainte n'était pas critique.

Dans les problèmes réels possédant une solution, elle est souvent unique. En effet, une solution multiple implique une fonction de coût parallèle à un sous-simplexe du simplexe initial. En pratique cette situation est peu probable. On la rencontre toutefois dans certains cas particuliers, par exemple si l'ensemble des contraintes est peu quantifié. Ce peut être le cas dans le cas des problèmes en nombre entiers, comme nous le verrons plus loin.

3.1.2 Solutions multiples

Dans le cas où la solution n'est pas unique, il faut pouvoir remarquer ce fait et probablement trouver *toutes* les solutions, c'est-à-dire toutes les combinaisons des variables qui donnent un coût optimal constant.

Pour illustrer ce point, reprenons l'exemple de l'ébénisterie en changeant un paramètre : on suppose qu'une table rapporte 35 euros et non seulement 30. Le reste est inchangé.

La formulation est la même que précédemment (sauf la fonction de coût). On prend comme base de départ la même que précédemment.

Première itération On a initialement VB=$\{x_4, x_5, x_6, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 48 \\ 20 \\ 8 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 8 & 6 & 1 \\ 4 & 6 & 1.5 \\ 2 & 1.5 & 0.5 \\ 0 & 1.0 & 0 \end{bmatrix}$$

Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cc} & \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ C_e^T = & [\text{-60} \quad \text{-35} \quad \text{-20}] \end{array}$$

Donc x_1 (le min) entre dans la base.

$P = B^{-1}A_1 = [\ 8 \quad 4 \quad 4 \quad 0\ ]^T$

ratios :

$$\begin{array}{ccccc} & x_4 & x_5 & x_6 & x_7 \\ \frac{\bar{b}}{P} = & 6 & 5 & 4 & \infty \end{array}$$

donc x_5 (le min dont P est positif) sort de la base.

Pour l'instant le déroulement de l'algorithme est le même que précédemment.

Deuxième itération On a maintenant VB=$\{x_4, x_5, x_1, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 0 & 8 & 0 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 16 \\ 4 \\ 4 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 6 & 1 & 0 \\ 2 & 1.5 & 0 \\ 1.5 & 0.5 & 1 \\ 1 & 0.0 & 0 \end{bmatrix}$$

Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cc} & \begin{array}{ccc} x_2 & x_3 & x_6 \end{array} \\ C_e^T = & [\ 10 \quad \text{-5} \quad 30\ ] \end{array}$$

Donc x_3 (le min) entre dans la base.

$P = B^{-1}A_3 = [\ -1 \quad 0.5 \quad 0.25 \quad 0\ ]^T$

ratios :

$$\begin{array}{ccccc} & x_4 & x_5 & x_1 & x_7 \\ \frac{\bar{b}}{P} = & \text{-16} & 8 & 16 & \infty \end{array}$$

donc x_5 (le min dont P est positif) sort de la base.

3^e^itération On a maintenant VB=$\{x_4, x_3, x_1, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 1.0 & 8 & 0 \\ 0 & 1.5 & 4 & 0 \\ 0 & 0.5 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 24 \\ 8 \\ 2 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 6 & 1 & 0 \\ 2 & 1 & 0 \\ 1.5 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cc} & \begin{array}{ccc} x_2 & x_5 & x_6 \end{array} \\ C_e^T = & [\begin{array}{ccc} 0 & 10 & 10 \end{array}] \end{array}$$

On a trouvé *un* optimum, mais x_2 est a zéro. Ceci indique une solution non unique, car on peut faire entrer x_2 dans la base à coût constant.

Donc x_2 (le min) entre dans la base.

3

3eitération, suite On poursuit le calcul normalement :

$P = B^{-1}A_3 = [\ -2 \quad -2 \quad 1.25 \quad 1\ ]^T$

ratios :

$$\begin{array}{ccccc} & x_4 & x_3 & x_1 & x_7 \\ \frac{\bar{b}}{P} = & -12 & -4 & 1.6 & 5 \end{array}$$

donc x_1 (le min dont P est positif) sort de la base.

4eitération On a maintenant VB=$\{x_4, x_3, x_2, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 1.0 & 6 & 0 \\ 0 & 1.5 & 2 & 0 \\ 0 & 0.5 & 1.5 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 27.2 \\ 11.2 \\ 1.6 \\ 3.4 \end{bmatrix}, E = \begin{bmatrix} 8 & 0 & 0 \\ 4 & 1 & 0 \\ 2 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cc} & \begin{array}{ccc} x_1 & x_5 & x_6 \end{array} \\ C_e^T = & [\begin{array}{ccc} 0 & 10 & 10 \end{array}] \end{array}$$

On a trouvé *un* optimum, mais x_1 est a zéro. Ceci indique une solution non unique, car on peut faire entrer x_1 dans la base à coût constant.

4eitération, suite et fin Donc si on fait de nouveau entrer x_1 (le min) entre dans la base.

$P = B^{-1}A_3 = [\ -2 \quad -2 \quad 1.25 \quad 1\ ]^T$

ratios :

$$\begin{array}{ccccc} & x_4 & x_3 & x_2 & x_7 \\ \frac{\bar{b}}{P} = & 17 & 7 & 2 & -4.25 \end{array}$$

donc x_2 (le min dont P est positif) sort de la base. On retrouve la même situation qu'à l'itération 3. On a découvert un cycle.

Solution non-unique La solution est constituée des variables de base réalisables possibles. Ici

$$\text{VB}_1 = \{x_4, x_3, x_1, x_7\}\text{VB}_2 = \{x_4, x_3, x_2, x_7\}$$

et de toutes leurs combinaisons linéaires intermédiaires.

Les valeurs de ces variables sont données par

$$\bar{b}_1 = \{24, 8, 2, 5\}\bar{b}_2 = \{27.2, 11.2, 1.6, 3.4\}$$

respectivement. Toutes les autres valeurs sont à 0.

En ne comptant que les variables entrant dans le coût, les deux points optimaux extrêmes sont :

$$e_1 = \begin{bmatrix} x_1 = & 2 \\ x_2 = & 0 \\ x_3 = & 8 \end{bmatrix}, e_2 = \begin{bmatrix} x_1 = & 0 \\ x_2 = & 1.6 \\ x_3 = & 11.2 \end{bmatrix}$$

Solutions intermédiaires Toutes les solutions intermédiaires sont données par :

$$\begin{bmatrix} x_1 = & 2c \\ x_2 = & 1.6 - 1.6c \\ x_3 = & 11.2 - 3.2c \end{bmatrix}$$

avec $0 \leq c \leq 1$.

La fonction de coût vaut donc :

$$z = -60x_1 - 30x_2 - 20x_3 = -280$$

Elle est constante pour toutes ces solutions.

Pour des problèmes plus complexes, on peut avoir plusieurs variables à zéro dans P, et non plus une seule. L'ensemble des solution est alors l'espace vectoriel convexe induit par les solutions extrêmes (un sous-simplexe du simplexe originel). Pour les trouver il faut réaliser toute les substitutions possibles autorisées à coût constant, c'est-à-dire toutes les permutations possible de ces variables à zéro dans P..

3.1.3 Solution non bornée

Un problème de programmation linéaire peut ne pas avoir de solution bornée. Pour illustrer ce cas, nous allons prendre un autre exemple, celui d'une boulangerie.

- Soit une boulangerie qui fabrique des petits pains ordinaires et des pains campagnards ;
- Les pains ordinaires se vendent pour 36 centimes et les pains campagnards 30 centimes ;
- Un pain ordinaire nécessite une dose de levure et 60g de farine, un pain campagnard une dose de levure et 50g de farine.
- La boulangerie possède pour le moment 5 doses de levure et 100g de farine.
- La levure coûte 3 centime la dose, et la farine 4 centimes les 10g.
- Maximisez le profit de la boulangerie.

Formulation

On pose les variables suivantes :

- x_1 = nombre de pains ordinaires produits
- x_2 = nombre de pains campagnards produits
- x_3 = nombre de doses de levure
- x_4 = farine consommée par quantité de 10g.

Les autres caractéristiques du problème sont les suivantes :

- Revenus $= 36x_1 + 30x_2$, coûts $= 3x_3 + 4x_4$.
- Objectif $= \max z = 36x_1 + 30x_2 - 3x3 - 4x_4$
- Contraintes 1 : $x_1 + x_2 \leq 5 + x_3$

– Contraintes 2 : $6x_1 + 5x_2 \leq 10 + x_4$

Mise sous forme standard On introduit deux variables d'écart pour les contraintes : x_5, x_6, toutes deux positives.

La forme standard est la suivante :

$$\begin{array}{lcrcrcrcrcrcrl}
\min & z = & -36x_1 & - & 30x_2 & + & 3x_3 & + & 4x_4 & & & & & \\
& & x_1 & + & x_2 & - & x_3 & & & + & x_5 & & & = 5 \\
& & 6x_1 & + & 5x_2 & & & - & x_4 & & & + & x_6 & = 10
\end{array}$$

Résolution

Une base naturelle de départ est la base VB=$\{x_5, x_6\}$, on procède comme d'habitude maintenant :

Première itération VB=$\{x_5, x_6\}$; VHB=$\{x_1, x_2, x_3, x_4\}$

$\bar{b} = \begin{bmatrix} 5 & 10 \end{bmatrix}$

Coûts réduits = $\begin{bmatrix} -36 & -30 & 3 & 4 \end{bmatrix}$ on va donc faire rentrer x_1.

$P = \begin{bmatrix} 1 & 6 \end{bmatrix}$

Ratios = $\begin{bmatrix} 5 & 1.66667 \end{bmatrix}$ on va donc faire sortir x_6.

Deuxième itération VB=$\{x_5, x_1\}$; VHB=$\{x_2, x_3, x_4, x_6\}$

$\bar{b} = \begin{bmatrix} 3.333 & 1.667 \end{bmatrix}$

Coûts réduits = $\begin{bmatrix} 0 & 3 & -2 & 6 \end{bmatrix}$ on va donc faire rentrer x_4.

$P = \begin{bmatrix} 0.167 & -0.167 \end{bmatrix}$

Ratios = $\begin{bmatrix} 20 & -10 \end{bmatrix}$ on va donc faire sortir x_5.

3$^{\text{e}}$itération VB=$\{x_4, x_1\}$; VHB=$\{x_2, x_3, x_5, x_6\}$

$\bar{b} = \begin{bmatrix} 20 & 5 \end{bmatrix}$

Coûts réduits = $\begin{bmatrix} 2 & -9 & 12 & 4 \end{bmatrix}$ on va donc faire rentrer x_3.

$P = \begin{bmatrix} 0.167 & -0.167 \end{bmatrix}$

Ratios = $\begin{bmatrix} -6 & -1 \end{bmatrix}$

Tous les ratios sont négatifs, on peut donc rendre la solution aussi négative que l'on veut. La solution n'est pas bornée.

Solution pour la boulangerie Avec la dernière solution de base réalisable, le système s'écrit :

$$\begin{array}{lrl}
x_4 & -6x_3 & = 20 \\
x_1 & -x_3 & = 5
\end{array}$$

Donc en augmentant x_3 de façon arbitraire, on fait aussi croître x_4 et x_1 arbitrairement, et donc on peut réduire z sans fin, tout en obéissant à toutes les contraintes. On voit bien que la solution n'est pas bornée.

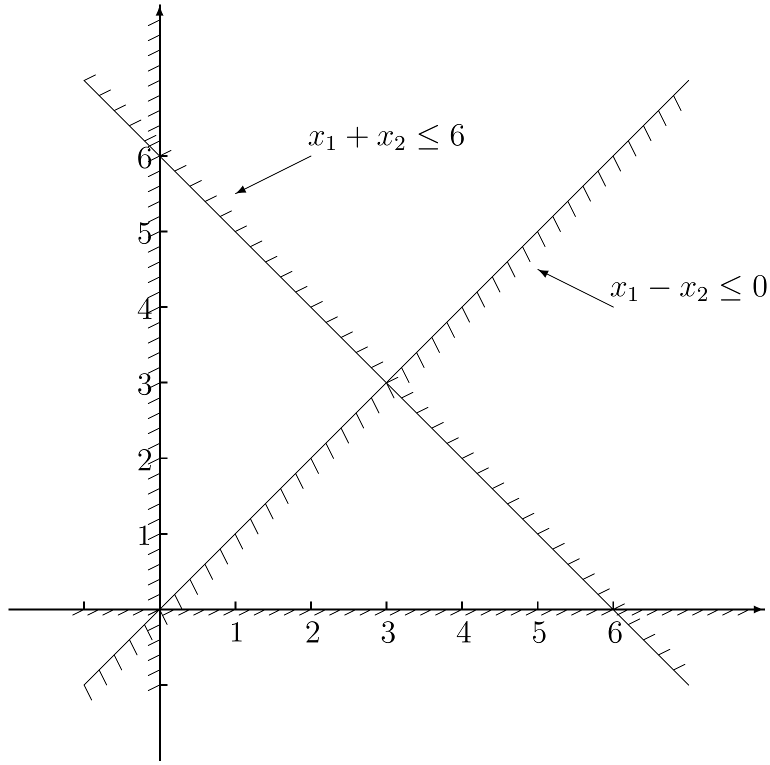


FIG. 3.1 – Exemple de situation dégénérée

3.1.4 Solution dégénérée

Soit le problème suivant

$$
\begin{array}{lrrrrrrrrr}
\min & z = & -5x_1 & - & 2x_2 & & & & & \\
 & & x_1 & + & x_2 & + & x_3 & & & = 6 \\
 & & x_1 & - & x_2 & & & + & x_4 & = 0
\end{array}
$$

3.1.5 Pas de solution

On peut facilement proposer un problème de programmation linéaire sans solution, par exemple :

$$
\begin{array}{rcl}
x_1 & \geq & 2 \\
x_1 & \leq & 1
\end{array}
$$

Clairement, quelle que soit la fonction de coût, ce système ne peut pas avoir de solution. En réalité un programme linéaire un peu complexe sans solution n'est jamais aussi trivial.

Un problème de PL sans solution ne possède aucune solution de base réalisable, en particulier on ne peut pas trouver de SBR initiale.

Or d'une manière générale, un problème de PL peut ne pas avoir de SBR initiale triviale, contrairement aux problèmes que nous avons vus jusqu'à présent sous forme d'illustration.

3.2 Initialisation de l'algorithme

3.2.1 Méthode avec les grands M dans la fonction de coût

3.2.2 Méthode en deux temps en minimisant d'abord les variables auxiliaires

3.3 Dualité

3.3.1 Problème primal/problème dual

3.3.2 Problème vendeur-consommateur

3.3.3 Interprétation

Unicité de la solution dans les deux
Multiplicité de la solution dans les deux
solution non-bornée / solution non-existante

3.3.4 Utilité de la dualité

Une des formes, soit primal soit duale est généralement plus efficace.

3.3.5 Passer de la solution du primal au dual et vice-versa

Quand elle est unique (solution finale)
Au cours de l'algorithme

3.3.6 Algorithme primal-dual ?

Permet d'encadrer l'optimum rapidement.

3.4 Efficacité de l'algorithme

Un exemple où le simplexe doit explorer tous les sommets avant de trouver l'optimum.

Deuxième partie

Exemples

Bibliographie

[1] J. L. Casti. *Five Golden Rules*. J. Wiley and Sons, new edition, 1997. ISBN 0-471-19337-2.

[2] G.B. Dantzig. Application of the simplex method to a transportation problem. In Tj.C. Koopmans, editor, *Activity Analysis of Production and Allocation*, Proceedings Conference on Linear Programming, Chicago, Illinois, 1949.

[3] M. Minoux. *Programmation Mathématique, Théorie et Algorithmes*, volume 1 of *Collection technique et scientifique des Télécommunications*. Dunod, Paris, 1983. ISBN 2-04-015487-6.

Table des figures

Index

Introduction à l'optimisation

Programmation linéaire

Hugues Talbot

Laboratoire A2SI

17 mars 2008

Plan

Intervenants

- Hugues Talbot : Introduction, programmation linéaire, simplexe.
- Hugues Talbot : Programmation en nombres entiers : coupes de Gomorry, B&B.
- Yskandar Hamam : Problèmes de flots.

Contenu du cours

- Introduction au problème, motivations
- Programmation linéaire
- Modélisation
- Méthode du simplexe
- Problèmes duaux/primaux
- Liens avec les graphes
- Programmation en nombre entiers
- Introduction aux autres méthodes d'optimisation

Emplois du temps et contrôle

- Partie Talbot
- 12h cours, 4h TD
- Partie Hamam
- 4h cours, 4h TD
- 1 TP de 3h
- Contrôle écrit, 2h, avec documents.

Exemples de problèmes

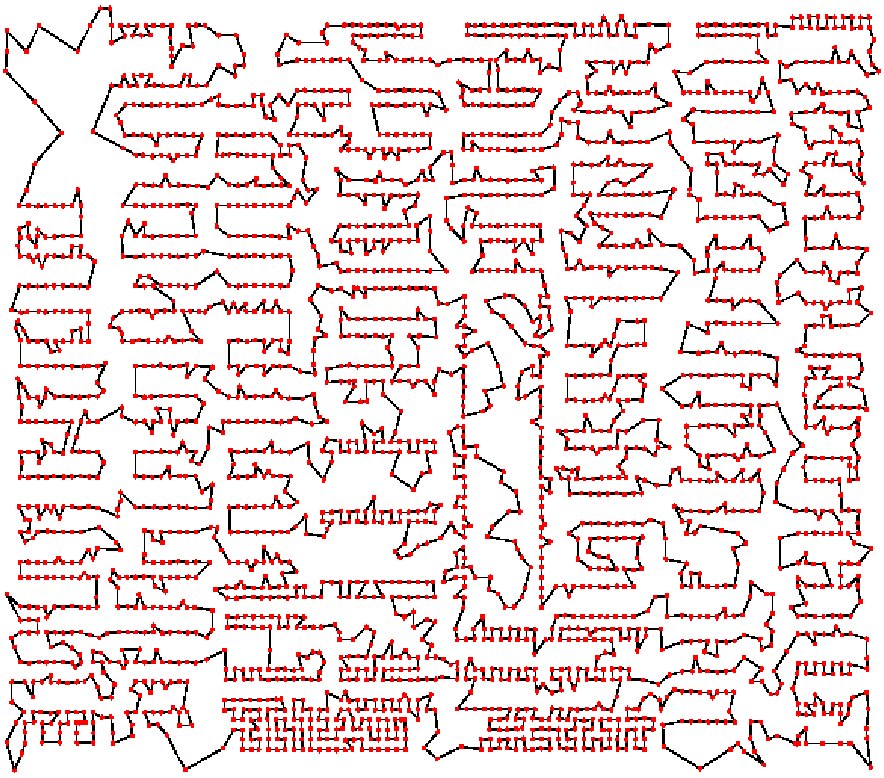
- Problème du voyageur de commerce (Travelling Salesman Problem – TSP) ;
- Autres problèmes NP-difficiles : knapsack, etc.
- Transport, consommation, réseaux, etc.

TSP

- Soit un voyageur de commerce devant visiter N villes
- Quel parcours minimise la distance totale parcourue ?
- Le problème décisionnel associé : “soit un réseau de N villes, existe t-il un parcours de longueur inférieure à K kilomètre ?” appartient à NP.
- Il n’existe pas de méthode connue fondamentalement meilleure que celle qui consiste à tout essayer :
  - 2 villes : 1 parcours (équivalents)
  - 5 villes : 60 parcours
  - n villes : $n!/2$ parcours.

Exemples de TSP - PCB routing, 3038 trous

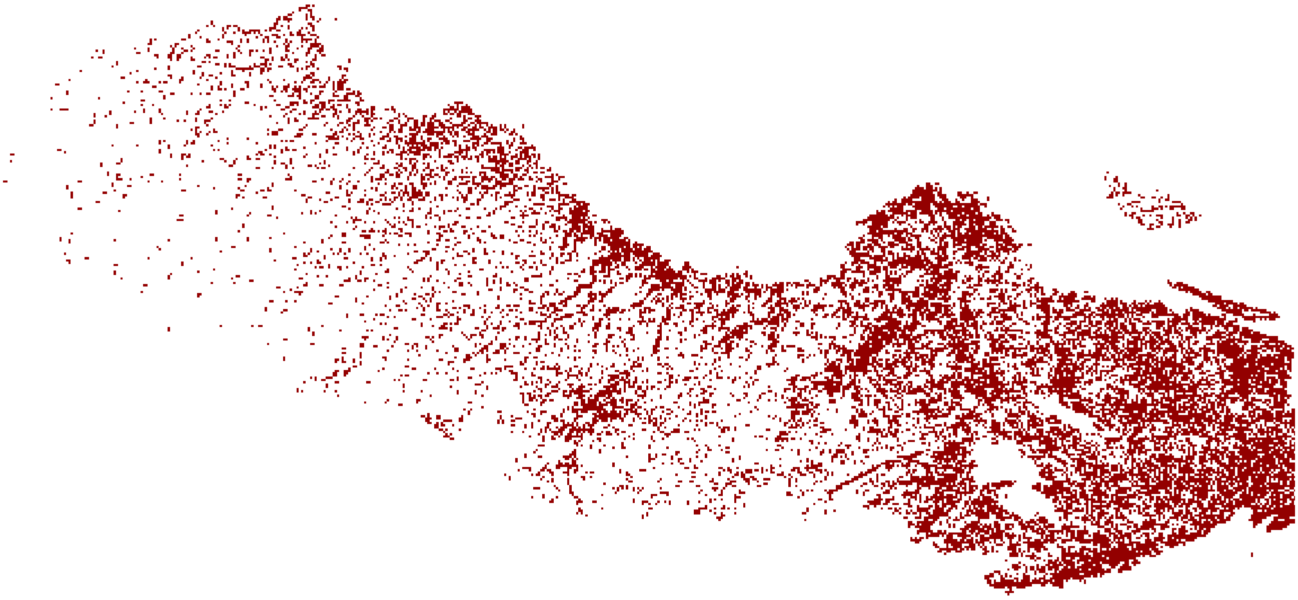
Exemples de TSP - PCB routing, 3038 trous



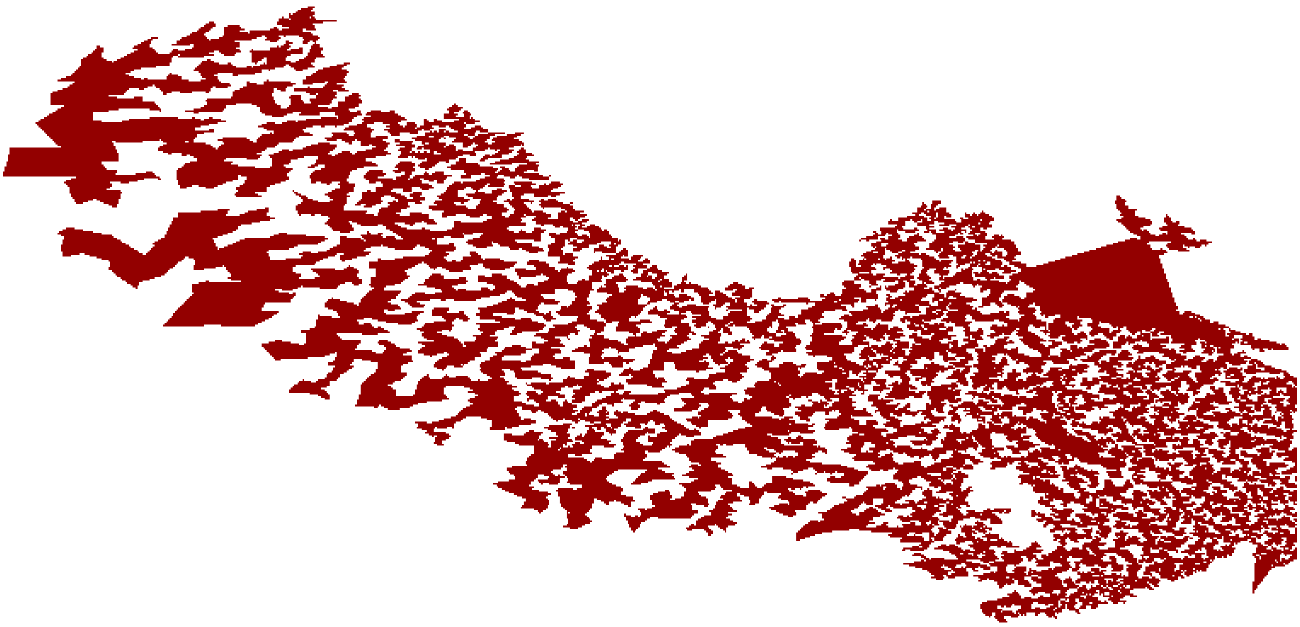
Histoire du TSP

Année	Équipe	Taille
1954	Dantzig, Fulkerson, Johnson	49
1971	Held, Karp	64
1977	Grötschel	120
1987	Padberg, Rinaldi	2392
2004	Applegate et al.	24978

Record actuel : tour de Suède



Record actuel : tour de Suède



Allocation de ressources disponibles

- Étant donné un pb, allouer les ressources au mieux.
- Très courant : argent, matériaux, personnel, temps, etc.
- Certains problèmes ont une solution optimale calculable, d'autres non.
- Une classe importante de problèmes non triviaux à solution optimale calculable sont les problèmes de *programmation linéaire*.

Nourriture pour chiens

- Soit une entreprise de nourriture pour chiens, qui fabrique 2 types de granulés : le Wag-Tail (W) et le Bark-Mad (B).
- Chacun des types utilise un mélange de légumes, boeuf et poisson, dans les proportions suivantes :

Ingrédient	Qté totale	Qté dans B	Qté dans W
Légumes	1400 kg	4 kg	4 kg
Poisson	1800 kg	6 kg	3 kg
Boeuf	1800 kg	2 kg	6 kg

- On suppose que la companie opère un bénefice de 12 Euros sur chaque paquet de B et de 8 euros sur ceux de W.
- Comment l'entreprise peut-elle faire pour maximiser son profit ?

Formulation

- On note par B le nombre de paquet de Bark-Mad produits, et par W le nombre de paquets de Wag-Tail.
- De la table précédente on déduit que la quantité totale de légumes consommé sera de $4W + 4B$ kg, mais qu'on ne peut en aucun cas utiliser plus de 1400 kg de légumes. Donc :

$$4B + 4W \leq 1400 \qquad (1)$$

Formulation II - contraintes

- De même :

$$6B + 3W \leq 1800 \qquad (2)$$

- et

$$2B + 6W \leq 1800 \qquad (3)$$

- finalement B et W sont tous deux positifs.

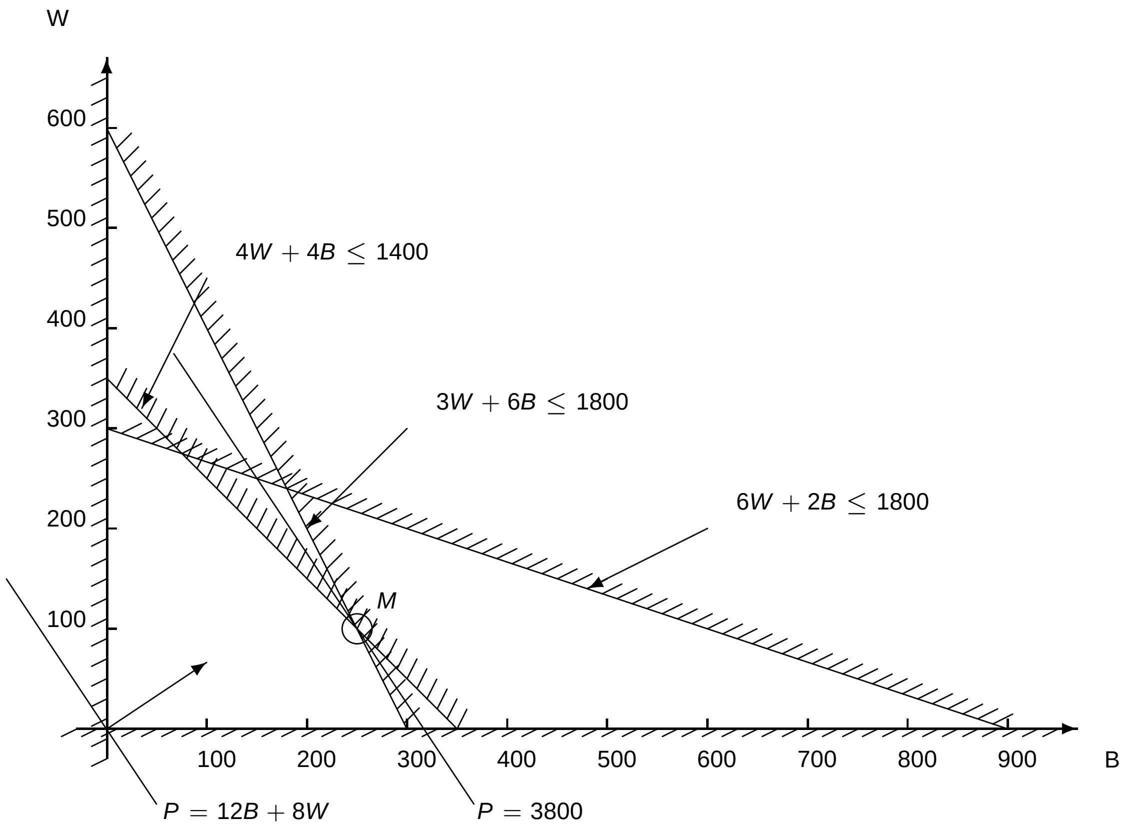
Formulation III - objectif

- Le profit total est :

$$P = 12B + 8W \tag{4}$$

- On cherche à maximiser P
- Pour un problème de cette nature : un dessin

Nourriture pour chiens, forme graphique



Solution

- On déplace une ligne à partir de l'origine parallèlement à elle-même d'équation égale à P.
- On s'arrête lorsqu'on maximise P en satisfaisant les contraintes
- Ici la solution est $B = (250, 100)$ – En ce point 2 ressources au moins sont totalement utilisées.

Vocabulaire & notes

- Un solution est réalisable si elle satisfait les contraintes ;
- Ensemble des solution réalisable forme un polygone (polyèdre, polytope) convexe ;
- La solution optimale est située sur une extrémité du polygone ;
- Pour trouver la solution il suffit d'explorer les points extrêmes du polygone.
- On note qu'on passe d'un ensemble infini de solution possible à un ensemble fini.

Solutions possibles à un problème de PL

1. La solution est unique (cas précédent) ;
2. La solution n'est pas forcément unique ;
3. Il n'y a pas forcément de solution ;
4. Il peut ne pas y avoir de solution bornée.

Historique

- En 1947, G. B. Dantzig travaillait come civil au Pentagone en tant qu'expert mathématicien ;
- Il avait à résoudre des problèmes de planning (avion, personnel, etc) ;
- Après discussion avec l'économiste T. Koopmans, il se rendit compte qu'il n'existait aucune méthodologie pour trouver une solution à ces problèmes.
- Dantzig s'est donc mis à en chercher une.

Idée de base

- En plus de 3 dimensions, l'objet géométrique équivalent à un polygone est appelé un *polytope*, et en topologie algébrique, un *simplexe*.
- Si la solution est sur une des arêtes du polytope, on peut partir d'une solution réalisable, parcourir les arêtes jusqu'à trouver un sommet optimal.
- Dantzig pensait que ça ne serait pas efficace, mais empiriquement on trouve la solution optimale en autant de déplacement que de dimensions dans l'espace des solutions (donc souvent très peu).

La méthode du simplexe

1. Partir d'une solution réalisable et calculer la fonction objectif en ce point ;
2. Trouver les arêtes de frontières de l'ensemble réalisable passant par ce point, calculer si la fonction objectif s'améliore en se déplaçant le long de cette arête ;
3. Se déplacer le long de l'arête donnant la plus grande augmentation ;
4. Répéter (2) et (3) jusqu'à ce que la fonction objectif n'augmente plus. On a trouvé la solution optimale.

Sur l'exemple de la nourriture pour chiens

- On élimine les inégalités par 3 variables artificielles s_i.
- On maximize $P = 12B + 8W$
- Avec les contraintes :

$$4B + 4W + s_1 = 1400 \quad (5)$$
$$6B + 3W + s_2 = 1800 \quad (6)$$
$$2B + 6W + s_3 = 1800 \quad (7)$$

Terminologie

- Ensemble des valeurs B, W et s_i = *solution réalisable*
- Une solution avec m équations et m inconnues avec certaines de ces variables à 0 est une solution de base
- L'ensemble des m variables non à zero forme une *base*.

Choix initial

- Pour former une solution de base, on doit choisir 3 variables et mettre les autres à 0.
- On peut poser $B = W = 0$ et résoudre pour les s_i
- Cela donne :

$$
\begin{aligned}
s_1 &= 1400 - 4B - 4W && (8)\\
s_2 &= 1800 - 6B - 3W && (9)\\
s_3 &= 1800 - 2B - 6W && (10)\\
P &= 12B + 8W && (11)
\end{aligned}
$$

- Géométriquement, cette solution est réalisable et correspond à l'origine. Malheureusement pour ce choix, $P = 0$, ce qui n'est pas optimal.

Première itération

- Pour augmenter P, on peut choisir d'augmenter B ou W. Comme B offre le plus grand gain, on choisit cette variable. W reste à 0.
- On introduit donc B dans la base, mais B ne peut pas être plus grand que 300, sinon s_2 deviendrait négative. On choisit donc $B = 300$, ce qui induit $s_2 = 0$.
- On réexprime le système avec les variables de base en fonction des variables hors-base :
la deuxième équation devient :

$$
\begin{aligned}
s_2 &= 1800 - 6B - 3W\\
-6B &= 1800 - s_2 - 3W\\
B &= 300 - \frac{1}{6}s_2 - \frac{1}{2}W
\end{aligned}
$$

Résultat de la première itération

- En substituant l'identité de B dans la première equation :

$$s_1 = 1400 - 4(300 - \frac{1}{6}s_2 - \frac{1}{2}W) - 4W = 200 + \frac{2}{3}s_2 - 2W$$

- En faisant de même pour la troisième équation et la fonction objectif, on obtient :

$$B = 300 - \frac{1}{6}s_2 - \frac{1}{2}W \quad (12)$$
$$s_1 = 200 + \frac{2}{3}s_2 - 2W \quad (13)$$
$$s_3 = 1200 + \frac{1}{3}s_2 - 5W \quad (14)$$
$$P = 3600 - 2s_2 + 2W \quad (15)$$

- Maintenant, avec s_2 et W à zero, le profit est de 3600.

Seconde itération

- On peut encore faire mieux en augmentant W, mais pour que s_1 reste positive, W ne peut pas être plus grand que 100.
- On introduit donc W dans la base, avec $W = 100$ ce qui induit $s_1 = 0$.
- On réexprime le système avec ces données :

$$W = 100 - \frac{1}{2}s_1 + \frac{1}{3}s_2 \quad (16)$$
$$B = 250 + \frac{1}{4}s_1 - \frac{1}{3}s_2 \quad (17)$$
$$s_3 = 700 + \frac{5}{2}s_1 - \frac{4}{3}s_2 \quad (18)$$
$$P = 3800 - s_1 - \frac{4}{3}s_2 \quad (19)$$

- Maintenant le profit est de 3800.

Optimum

- On ne peut plus augmenter le profit P, car augmenter s_1 ou s_2 diminue P.
- On peut interpreter le fait que s_1 (légumes) et s_2 (poisson) soient à 0 par le fait que ces ressources sont complètement consommées, ce qui n'est pas le cas de s_3 (boeuf).

Résumé

- La méthode du simplexe permet d'optimiser les problèmes de programmation linéaire.
- La méthode a une interprétation géométrique simple, et donne lieu à un algorithme.
- En pratique, quelques itérations suffisent pour obtenir l'optimum quand il existe.
- Nous verrons dans la suite du cours l'algorithme en détail, y compris les cas limites.

Algorithme du simplexe

Une solution à la programmation linéaire

Hugues Talbot

Laboratoire A2SI

18 mars 2008

Plan

Matrices, inverses etc

- Il est nécessaire de maîtriser un minimum d'algèbre linéaire : matrices (addition, multiplication etc), inverses etc.
- Pour les exercices, TD, TPs et examen, on peut vous demander d'inverser à la main une matrice 3×3.
- Un cours complet d'algèbre linéaire : `http://joshua.smcvt.edu/linearalgebra/` : 440 pages, libre, avec toutes les preuves et la solution de tous les exercices.

Algèbre linéaire Algorithme du simplexe Résumé

Exemple - fabrique de ceintures

- Une usine de ceinture en fabrique de 2 sortes : luxe et standard
- Chaque type demande $1m^2$ de cuir
- Une ceinture standard demande 1h de travail
- Une centure de luxe 2h
- Chaque semaine, on dispose de $40m^2$ de cuir et de 60h de travail.
- Chaque ceinture standard rapport 3 Euros
- Chaque ceinture de luxe 4 Euros.
- Maximiser le profit.

Formulation

- $x_1 =$ nombre de ceintures de luxe produites par semaine
- $x_2 =$ nombre de ceintures standard produites par semaine
- Maximiser $z = 4x_1 + 3x_2$, avec

$$x_1 + x_2 \leq 40 \qquad \text{contrainte sur le cuir} \qquad (1)$$
$$2x_1 + x_2 \leq 60 \qquad \text{contrainte sur le travail} \qquad (2)$$
$$x_1, x_2 \geq 0 \qquad \text{contrainte de signe} \qquad (3)$$

Conversion en forme standard

- On souhaite convertir toute les inégalités en égalités.
- Pour chaque variable $\leq$ on définit une variable de "manque" s_i. Toutes les s_i sont positives. Ici

$$s_1 = 40 - x_1 - x_2 \qquad (4)$$
$$s_2 = 60 - 2x_1 - x_2 \qquad (5)$$

- Le problème s'écrit maintenant sous la *forme standard* : Maximiser z, avec :

$$\begin{array}{ccccccccccc} z & = & 4x_1 & + & 3x_2 & & & & & & \\ & & x_1 & + & x_2 & + & s_1 & & & = & 40 \\ & & 2x_1 & + & x_2 & & & + & s_2 & = & 60 \\ & & & & x_1, & x_2, & s_1, & & s_2 & \geq & 0 \end{array}$$

Problème du régime

- On veut suivre un régime qui impose de manger un élément des 4 groupes de base : chocolat, crème glacée, soda et gâteau.
- Une barre de chocolat coûte 50 centimes, une part de crème glacée 20 centimes, chaque bouteille de soda 30 centimes et une part de gâteau 80 centimes.
- Chaque jour je dois ingérer 500 calories, 60g de chocolat, 100g de sucre et 80g de lipides.
- Le contenu nutritionnel de chaque type de nourriture est donné ainsi

	Cal.	Choc. (g)	Sucr. (g)	Lip. (g)
barre chocolat	400	30	20	20
crème glacée	200	20	20	40
cola	150	0	40	10
gâteau	500	0	40	50

Formulation

- On veut minimizer le coût du régime.
- Combien de variables ?
- Exprimer la fonction objectif
- Exprimer les contraintes
- Mettre sous forme standard

Formulation – régime

- Objectif = min $z = 50x_1 + 20x_2 + 30x_3 + 80x_4$
- Total calories = $400x_1 + 200x_2 + 150x_3 + 500x_4 \geq 500$
- Total chocolat = $30x_1 + 20x_2 \geq 60$
- Total sucres = $20x_1 + 20x_2 + 40x_3 + 40x_4 \geq 100$
- Total lipides = $20x_1 + 40x_2 + 10x_3 + 50x_4 \geq 80$
- Finalement, tous les x_i sont positifs.

Formulation – forme standard

- Pour des contraintes $\geq$ on définit des variables d'excès e_i toutes positives.
- On obtient :

$$\begin{array}{rcrcrcrcrcrcrcrcrcrcr}
z & = & 50x_1 & + & 20x_2 & + & 30x_3 & + & 80x_4 & & & & & & & & & & \\
& & 400x_1 & + & 200x_2 & + & 150x_3 & + & 500x_4 & & -e1 & & & & & & & = & 500 \\
& & 30x_1 & + & 20x_2 & & & & & & & & -e2 & & & & & = & 60 \\
& & 20x_1 & + & 20x_2 & + & 40x_3 & + & 40x_4 & & & & & & -e3 & & & = & 100 \\
& & 20x_1 & + & 40x_2 & + & 10x_3 & + & 50x_4 & & & & & & & & -e4 & = & 80
\end{array}$$

avec x_i et e_i positifs

- Avec des contraintes mixtes (c-à-d $\leq$ et $\geq$) on a à la fois des variables s_i et e_i.
- Les variables s_i et e_i deviennent partie du système au même titre que les variable x_i.

Forme standard générale

- On suppose un problème de programmation linéaire avec m contraintes et n variables sous forme standard.
- Il a la forme suivante : maximiser (ou minimiser) z avec

$$\begin{array}{ccccccccccc} z & = & c_1x_1 & + & c_2x_2 & + & \dots & + & c_nx_n & & \\ & & a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ & & a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ & & \vdots & & \vdots & & & & \vdots & & \\ & & a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & = & b_m \end{array}$$

et $\forall i, x_i \geq 0$

Matrice principale

On définit :

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Note : $n \geq m$, sinon le système est à-priori sur-contraint.

Matrices des variables et contraintes

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}.$$

PL sous forme matricielle

Le programme linéaire s'écrit sous forme matricielle :

$$\min \text{ (ou max) } \mathbf{c}^T\mathbf{x} \tag{6}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{7}$$

$$\mathbf{x} \geq 0 \tag{8}$$

Variables de base

- On appelle Base une sous-matrice régulière de **A**. Il faut que la matrice $\mathbf{A}(m, n)$ soit de rang m.
- Une solution de base est obtenue en posant $n - m$ variables égales à 0, et en résolvant pour les m variables restantes, qui sont les variables de base (VB).
- Les $n - m$ variables à 0 sont les *variables hors base* (VHB).
- Des choix différents de VHB donnent lieu à des différentes solutions de base.

Représentation

x_b^t	x_e^t

c_b^t	c_e^t

B base	E colonnes hors base
m colonnes	$n - m$ colonnes

Représentation matricielle

- On a

$$\mathbf{A} = [\mathbf{BE}], \mathbf{x} = \begin{bmatrix} x_b \\ x_e \end{bmatrix}, \mathbf{c}^T = \begin{bmatrix} \mathbf{c}_b{}^T \mathbf{c}_e{}^T \end{bmatrix}$$

- Ce qui donne
  $z = \mathbf{c}^T\mathbf{x} = \mathbf{c}_b{}^T\mathbf{x}_b + \mathbf{c}_e{}^T\mathbf{x}_e$
  $\mathbf{Ax} = \mathbf{b} \rightarrow \mathbf{Bx}_b + \mathbf{Ex}_e = \mathbf{b}$
- Une solution de base est telle que

$$\mathbf{x}_e = 0 \tag{9}$$

$$\mathbf{Bx}_b = \mathbf{b} \tag{10}$$

$$\mathbf{x}_b = \mathbf{B}^{-1}\mathbf{b} \tag{11}$$

Exemple

- Soit le système suivant :

$$\begin{array}{ccccccc} x_1 & + & x_2 & & & = & 3 \\ & & -x_2 & + & x_3 & = & -1 \end{array}$$

- Si on pose VHB $= \{x_3\}$, alors VB $= \{x_1, x_2\}$. On résout

$$\begin{array}{ccccc} x_1 & + & x_2 & = & 3 \\ & & -x_2 & = & -1 \end{array}$$

  Ce qui donne $x_1 = 2$ et $x_2 = 1$.
- Certains choix de variables peuvent ne pas générer de solution de base.

Solutions de base réalisables

- Une solution de base est dite réalisable (SBR) si

$$\mathbf{x}_b = \mathbf{B}^{-1}\mathbf{b} \geq 0$$

- Si le vecteur $\mathbf{x}_b$ contient des termes nuls, on dira que cette solution est une solution de base dégénérée.

Exemple de SBR - 1

- Soit le problème :

$$\begin{array}{lrrrrrl}
\min & - & x_1 & - & 2x_2 & & \\
\text{avec} & & & & & & \\
 & & x_1 & + & 2x_2 & \leq & 4 \\
 & & 2x_1 & + & x_2 & \leq & 5 \\
 & & x_1 & , & x_2 & \geq & 0
\end{array}$$

- Sous forme standard, on a

$$\begin{array}{lrrrrrrrrrl}
\min & - & x_1 & - & 2x_2 & & & & & & \\
\text{avec} & & & & & & & & & & \\
 & & x_1 & + & 2x_2 & + & x_3 & & & = & 4 \\
 & & 2x_1 & + & x_2 & & & + & x_4 & = & 5 \\
 & & x_1 & , & x_2 & , & x_3 & , & x_4 & \geq & 0
\end{array}$$

Exemple de SBR - 2

- On peut essayer de constituer une base en utilisant l'ensemble B = $\{1, 3\}$,

$$\mathbf{B} = [\mathbf{A}_1\mathbf{A}_3] = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$$

$$\mathbf{E} = [\mathbf{A}_2\mathbf{A}_4] = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$$

$$\mathbf{x}_b = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}, \mathbf{x}_e = \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}$$

- L'inverse de **B** existe, donc **B** correspond à une base

$$\mathbf{B}^{-1} = \begin{bmatrix} 0 & 1/2 \\ 1 & -1/2 \end{bmatrix}$$

Exemple de SBR - 3

- La solution de base correspondante est donc :

$$\mathbf{x}_b = \mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 0 & 1/2 \\ 1 & -1/2 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 5/2 \\ 3/2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} > 0$$

- Cette solution est bien une SBR.

Théorèmes fondamentaux

Théorème

La région réalisable pour tout problème de programmation linéaire est un ensemble convexe. Si un PL possède une solution optimale, alors un point extrême de la région réalisable doit être optimal.

Théorème

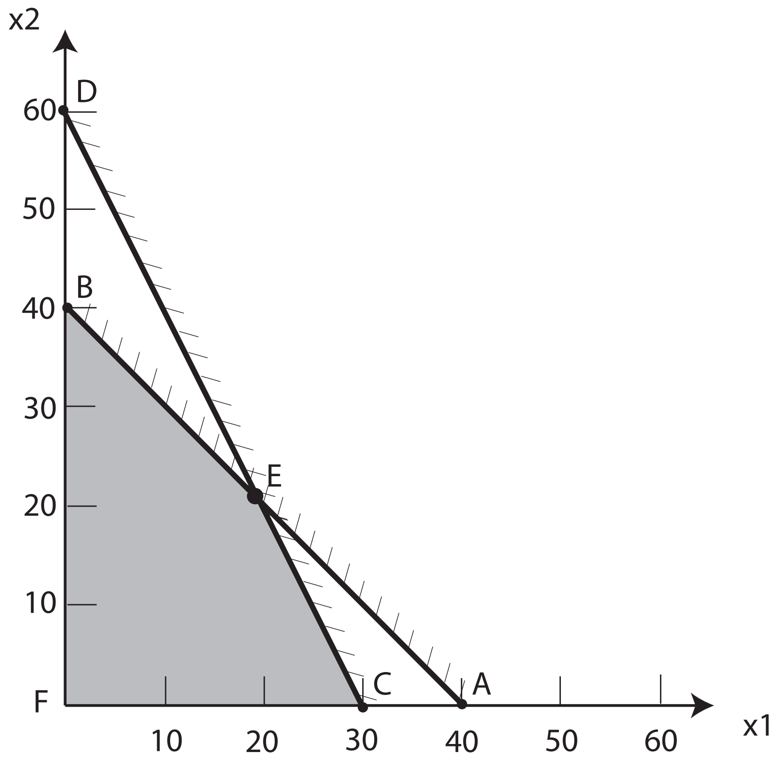
Pour tout LP, il existe un point extrême unique de la région réalisable qui correspond à chaque solution de base réalisable. Également, il existe au moins une SBR qui correspond à chaque point extrême de la région réalisable.

Illustration des théorèmes

On reprend l'exemple des ceintures de cuir, c-à-d maximiser z, avec :

$$
\begin{array}{rcrcrcrcrcr}
z & = & 4x_1 & + & 3x_2 & & & & & & \\
 & & x_1 & + & x_2 & + & s_1 & & & = & 40 \\
 & & 2x_1 & + & x_2 & & & + & s_2 & = & 60 \\
 & & & & x_1, & & x_2, & s_1, & s_2 & \geq & 0
\end{array}
$$

Ceintures de cuir - 1



Ceintures de cuir - 2

On a l'équivalence entre SBR et points extrêmes suivants :

Base	Hors-base	SBR	Point extrême
x_1, x_2	s_1, s_2	$s_1 = s_2 = 0, x_1 = x_2 = 20$	E
x_1, s_1	x_2, s_2	$x_2 = s_2 = 0, x_1 = 30, s_1 = 10$	C
x_1, s_2	x_2, s_1	$x_2 = s_1 = 0, x_1 = 40, s_2 = -20$	Non réalisable, $s_2 < 0$
x_2, s_1	x_1, s_2	$x_1 = s_2 = 0, s_1 = -20, x_2 = 60$	Non réalisable, $s_1 < 0$
x_2, s_2	x_1, s_1	$x_1 = s_1 = 0, x_2 = 40, s_2 = 20$	B
s_1, s_2	x_1, x_2	$x_1 = x_2 = 0, s_1 = 40, s_2 = 60$	F

Preuve

Soit **x** une SBR, de la forme $\mathbf{x} = \{x_1, x_2, \ldots, x_m, 0, 0, \ldots, 0]^T$. Si **x** n'est pas un point extrême, il existe 2 points (solutions) α et β différents de **x** et un scalaire λ tels que :

$$\mathbf{x} = \lambda\alpha + (1-\lambda)\beta, 0 < \lambda < 1$$

soit

$$\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_m, \alpha_{m+1}, \ldots, \alpha_n]^T = \begin{bmatrix} \alpha_b \\ \alpha_e \end{bmatrix}$$

$$\beta = [\beta_1, \beta_2, \ldots, \beta_m, \beta_{m+1}, \ldots, \beta_n]^T = \begin{bmatrix} \beta_b \\ \beta_e \end{bmatrix}$$

Alors

$$\lambda\alpha_i + (1-\lambda)\beta_i = 0 \forall i \in [m+1, n]$$

Puisque $\lambda > 0, (1-\lambda) > 0, \alpha_i > 0 et \beta_i > 0$, on a $\alpha_i = \beta_i = 0$, soit $\mathbf{x} = \alpha = \beta$, contradiction.

Nombre de solutions possibles

- Le nombre de bases candidates est $C_n^m = \frac{n!}{(n-m)!m!}$. Toutes les candidates ne sont pas inversibles, donc on peut seulement dire que le nombre précédent est une borne supérieure.
- Une méthode basée sur l'exploration des points extrêmes est cependant non-polynomiale.
- L'expérience montre que pour un problème de n variables à m contraintes, la solution optimale est trouvée en moyenne en moins de $3m$ opérations.

SBR adjacentes

- Pour tout problème de PL, deux SBR sont adjacentes si leur ensembles de variables de base ont $m - 1$ variables de base en commun.

L'interprétation géométrique est que les 2 SBRs sont situées le long d'une même arête sur le polytope réalisable.

Description générale de l'algorithme

L'algorithme du simplexe pour une maximisation suit les étapes suivantes :

1. Trouver une SBR pour le PL, appelée la SBR initiale.
2. Déterminer si la SBR courante est optimale. Sinon, trouver une SBR adjacente qui possède une valeur z plus élevée.
3. Retourner au point (2) avec la nouvelle SBR comme SBR courante.

Les deux questions suivantes sont donc : comment détecter l'optimalité, et comment se déplacer.

Coûts réduits

- Pour une SBR, on peut écrire :

$$z = \mathbf{c_b}^T \mathbf{x}_b + \mathbf{c_e}^T \mathbf{x}_e$$

et

$$\mathbf{B}\mathbf{x}_b + \mathbf{E}\mathbf{x}_e = \mathbf{b}$$

- Donc

$$\mathbf{x}_b = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{E}\mathbf{x}_e)$$

- Par substitution

$$z = \mathbf{c_b}^T \mathbf{B}^{-1}\mathbf{b} + (\mathbf{c_e}^T - \mathbf{c_b}^T \mathbf{B}^{-1}\mathbf{E})\mathbf{x_e}$$

- On pose :

$$\bar{\mathbf{c}}_e^T = (\mathbf{c_e}^T - \mathbf{c_b}^T \mathbf{B}^{-1}\mathbf{E})$$

Coûts réduits - 2

- Pour cette SBR, $\mathbf{x}_e = 0$, mais ce 2ème terme correspond à l'augmentation du coût pour une augmentation des variables dans $\mathbf{x}_e$.
- Si tous les coûts sont négatifs (pour une maximisation), toute augmentation des variables de $\mathbf{x}_e$ diminuera la valeur de z, et donc la solution obtenue est optimale.
- Réciproquement pour une minimisation.
- On a donc répondu à la première question (test d'optimalité).

Exemple

- Prenons le cas des ceintures de cuir, avec comme SBR=$\{s_1, s_2\}$ et SHB=$\{x_1, x_2\}$.
- Comme dans ce cas, $\mathbf{c_b}^T = [00]$, on a $\bar{\mathbf{c}}_e^T = \mathbf{c_e}^T = [43]$
- On voit que pour augmenter z le plus efficacement, on doit faire entrer x_1 dans la base, car son coefficient est plus élevé.
- On doit encore décider quelle variable faire *sortir* de la base. Pour cela, on doit faire augmenter x_1 en gardant x_2 à zéro, puis voir quelle variable de base s'annule la première.
- Dans notre cas, s_2 s'annule la première (voir dessin). C'est donc celle qu'on doit faire sortir.
- Si on ne fait pas cela correctement, on risque de choisir une base non réalisable.

Amélioration d'une solution de base

- Pour une maximisation, si notre base est telle que $\bar{\mathbf{c}}_e^T$ ne soit pas strictement négative ou nulle, alors il existe une variable x_k de $\mathbf{x}_e$ telle que $\bar{c}_k > 0$. Une augmentation de x_k est donc susceptible d'améliorer z.
- **Changement de base** Si pour une variable x_k de $\mathbf{x}_e$, $\bar{c}_k > 0$, la solution peut-être améliorée en augmentant x_k.

$$\mathbf{x}_b = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{A}_k x_k - \mathbf{E}'\mathbf{x}'_e)$$

En fixant $\mathbf{x}'_e = 0$, et en variant x_k seulement :

$$\mathbf{x}_b = \mathbf{B}^{-1}(\mathbf{b} - A_k x_k) = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{A}_k x_k \tag{12}$$

$$\mathbf{x}_b = \bar{\mathbf{b}} - \mathbf{P}x_k \tag{13}$$

Augmentation

- Comme originellement x_k est nulle, on ne peut que l'augmenter. Il y a deux cas :
- **cas 1** $\forall i, P_i \leq 0$. En ce cas la solution est non-bornée. x_k tend vers $+\infty$ et z vers $-\infty$.
- **cas 2**, il y a 2 possibilités, pour chaque i :
  1. soit $P_i \leq 0$, $x_{bi} \geq 0$ pour tout $x_k \geq 0$, donc cas non critique : on ne peut pas utiliser cette variable.
  2. soit $P_i > 0$, donc $x_{bi} \leq 0$ pour $x_k \geq \bar{b}_i/P_i$, donc pour tout $P_i > 0$, il existe une valeur maximale de $x_k = \bar{b}_i/P_i$, permettant $x_b \geq 0$.
     On choisit la variable l telle que :

$$l = \min_{i/P_i>0} \left[\frac{\bar{b}_i}{P_i}\right]$$

En résumé

- On a présenté un algorithme utilisant l'algèbre linéaire à la place de l'intuition graphique.
- Il faut savoir modéliser un problème.
- Il faut comprendre l'algorithme du simplexe.
- Il faut être capable de le faire tourner sur des exemples simples.

Programmation linéaire – suite

Cas limites du simplexe

Hugues Talbot

Laboratoire A2SI

6 avril 2007

Plan

Application de l'algorithme du simplexe

- L'algorithme décrit ne fonctionne que pour une *minimisation* de z, pour maximiser z on doit minimiser $-z$.
- On doit trouver une base initiale réalisable. Ce n'est pas toujours évident.
- L'algorithme fournit une base réalisable et une valeur extrême d'une des variables. On doit trouver les autres en résolvant le système.
- On doit tenir compte des cas limites.

Solution optimale unique

- Une ébénisterie produit des bureaux, des tables et des chaises
- Chaque type de produit réclame du bois et deux types de travaux : mise en forme et finition, suivant le tableau :

Ressource	bureau	table	chaise
planches	8m	6m	1m
mise en forme	4h	2h	1.5h
finition	2h	1.5h	0.5h

- On dispose de 48m de planches, 20h de mise en forme et 8h de finition.
- On vend un bureau pour 60 Euros, une table pour 30 Euros et une chaise pour 20 Euros.
- La demande pour les chaises et les bureaux est illimitée, mais on ne pense vendre que 5 tables au plus.
- On veut maximiser le profit.

Formulation

- Variables : x_1 = nb. bureaux, x_2 = tables, x_3= chaises
- Objectif : $\max z = 60x_1 + 30x_2 + 20x_3$
- Contraintes :

$$\begin{array}{rcrcrcr} 8x_1 & + & 6x_2 & + & x_3 & \leq & 48 \\ 4x_1 & + & 2x_2 & + & 1.5x_3 & \leq & 20 \\ 2x_1 & + & 1.5x_2 & + & 0.5x_3 & \leq & 8 \\ & & x_2 & & & \leq & 5 \end{array}$$

$x_1, x_2, x_3 \geq 0$

Forme standard

Toutes les inégalités sont des $\leq$, donc les variables d'écart sont positives.

$$\begin{array}{llrcrcrcrcrcrcrr} \min & z = & -60x_1 & - & 30x_2 & - & 20x_3 & & & & & & & & & \\ & & 8x_1 & + & 6x_2 & + & x_3 & + & x_4 & & & & & & & = 48 \\ & & 4x_1 & + & 2x_2 & + & 1.5x_3 & & & + & x_5 & & & & & = 20 \\ & & 2x_1 & + & 1.5x_2 & + & 0.5x_3 & & & & & + & x_6 & & & = 8 \\ & & & & x_2 & & & & & & & & & + & x_7 & = 5 \end{array}$$

Première itération

- Au départ :

$$A = \begin{bmatrix} 8 & 6 & 1 & 1 & 0 & 0 & 0 \\ 4 & 2 & 1.5 & 0 & 1 & 0 & 0 \\ 2 & 1.5 & 0.5 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 48 \\ 20 \\ 8 \\ 5 \end{bmatrix}$$

$$C^T = \begin{bmatrix} -60 & -30 & -20 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Comme base initiale on choisit VB=$\{x_4, x_5, x_6, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 48 \\ 20 \\ 8 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 8 & 6.0 & 1.0 \\ 4 & 2.0 & 1.5 \\ 2 & 1.5 & 0.5 \\ 0 & 1.0 & 0.0 \end{bmatrix}$$

Première itération – échanges de variables

- Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cccc} & x_1 & x_2 & x_3 \\ C_e^T = & [\,-60 & -30 & -20\,] \end{array}$$

Donc x_1 (le min) entre dans la base.

- $P = B^{-1} A_1 = \begin{bmatrix} 8 & 4 & 2 & 0 \end{bmatrix}^T$
- ratios :

$$\begin{array}{ccccc} & x_4 & x_5 & x_6 & x_7 \\ \frac{\bar{b}}{P} = & 6 & 5 & 4 & \infty \end{array}$$

donc x_6 (le min) sort de la base.

Deuxième itération

- On a maintenant VB=$\{x_4, x_5, x_1, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 0 & 8 & 0 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 16 \\ 4 \\ 4 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 6 & 1 & 0 \\ 2 & 1.5 & 0 \\ 1.5 & 0.5 & 1 \\ 1 & 0.0 & 0 \end{bmatrix}$$

- Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cccc} & x_2 & x_3 & x_6 \\ C_e^T = & [\,15 & -5 & 30\,] \end{array}$$

  Donc x_3 (le min) entre dans la base.
- $P = B^{-1}A_3 = [\ -1 \quad 0.5 \quad 0.25 \quad 0\ ]^T$
- ratios :

$$\begin{array}{ccccc} & x_4 & x_5 & x_1 & x_7 \\ \frac{\bar{b}}{P} = & -16 & 8 & 16 & \infty \end{array}$$

  donc x_5 (le min dont P est positif) sort de la base.

Troisième itération

- On a maintenant VB=$\{x_4, x_3, x_1, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 1.0 & 8 & 0 \\ 0 & 1.5 & 4 & 0 \\ 0 & 0.5 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 24 \\ 8 \\ 2 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 6 & 1 & 0 \\ 2 & 1 & 0 \\ 1.5 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

- Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cccc} & x_2 & x_5 & x_6 \\ C_e^T = & [\,5 & 10 & 10\,] \end{array}$$

  On a trouvé l'optimum !

Solution

- La solution est constituée des variable de base réalisable. Ici
$$VB=\{x_4, x_3, x_1, x_7\}$$
- Les valeurs de ces variables sont données par
$$\bar{b} = \{24, 8, 2, 5\}$$
respectivement. Toutes les autres valeurs sont à 0.
- La fonction de coût vaut donc :
$$z = -60x_1 - 30x_2 - 20x_3 = -280$$

Solution optimale multiple

- On prend le même problème, mais cette fois-ci on suppose qu'une table rapporte 35 Euros au lieu de 30.
- Le reste est inchangé

Première itération (table=35 Euros)

- On a initialement VB=$\{x_4, x_5, x_6, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 48 \\ 20 \\ 8 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 8 & 6 & 1 \\ 4 & 6 & 1.5 \\ 2 & 1.5 & 0.5 \\ 0 & 1.0 & 0 \end{bmatrix}$$

- Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cc} & \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ C_e^T = & [\begin{array}{ccc} -60 & -35 & -20 \end{array}] \end{array}$$

Donc x_1 (le min) entre dans la base.

- $P = B^{-1}A_1 = [\ 8 \quad 4 \quad 4 \quad 0\ ]^T$
- ratios :

$$\begin{array}{ccccc} & x_4 & x_5 & x_6 & x_7 \\ \frac{\bar{b}}{P} = & 6 & 5 & 4 & \infty \end{array}$$

donc x_5 (le min dont P est positif) sort de la base.

Cas limites de la programmation linéaire
PL dégénérée et convergence

Deuxième itération (table à 35 Euros)

- On a maintenant VB=$\{x_4, x_5, x_1, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 0 & 8 & 0 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 16 \\ 4 \\ 4 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 6 & 1 & 0 \\ 2 & 1.5 & 0 \\ 1.5 & 0.5 & 1 \\ 1 & 0.0 & 0 \end{bmatrix}$$

- Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cc} & \begin{array}{ccc} x_2 & x_3 & x_6 \end{array} \\ C_e^T = & [\begin{array}{ccc} 10 & -5 & 30 \end{array}] \end{array}$$

Donc x_3 (le min) entre dans la base.

- $P = B^{-1}A_3 = [\ -1 \quad 0.5 \quad 0.25 \quad 0\ ]^T$
- ratios :

$$\begin{array}{ccccc} & x_4 & x_5 & x_1 & x_7 \\ \frac{\bar{b}}{P} = & -16 & 8 & 16 & \infty \end{array}$$

donc x_5 (le min dont P est positif) sort de la base.

Troisième itération (tables à 35 Euros)

- On a maintenant VB=$\{x_4, x_3, x_1, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 1.0 & 8 & 0 \\ 0 & 1.5 & 4 & 0 \\ 0 & 0.5 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 24 \\ 8 \\ 2 \\ 5 \end{bmatrix}, E = \begin{bmatrix} 6 & 1 & 0 \\ 2 & 1 & 0 \\ 1.5 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

- Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cccc} & x_2 & x_5 & x_6 \\ C_e^T = & [\,0 & 10 & 10\,] \end{array}$$

On a trouvé *un* optimum, mais x_2 est a zéro. Ceci indique une solution non unique, car on peut faire entrer x_2 dans la base à coût constant.
Donc x_2 (le min) entre dans la base.

Troisième itération, suite

- On poursuit le calcul normalement :
- $P = B^{-1} A_3 = [\ -2 \quad -2 \quad 1.25 \quad 1\ ]^T$
- ratios :

$$\begin{array}{ccccc} & x_4 & x_3 & x_1 & x_7 \\ \frac{\bar{b}}{P} = & -12 & -4 & 1.6 & 5 \end{array}$$

donc x_1 (le min dont P est positif) sort de la base.

Quatrième itération (tables à 35 Euros)

- On a maintenant VB=$\{x_4, x_3, x_2, x_7\}$ On a donc

$$B = \begin{bmatrix} 1 & 1.0 & 6 & 0 \\ 0 & 1.5 & 2 & 0 \\ 0 & 0.5 & 1.5 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \bar{b} = \begin{bmatrix} 27.2 \\ 11.2 \\ 1.6 \\ 3.4 \end{bmatrix}, E = \begin{bmatrix} 8 & 0 & 0 \\ 4 & 1 & 0 \\ 2 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

- Coûts réduits $\bar{C}_e^T = C_e^T - C_b^T B^{-1} E$

$$\begin{array}{cc} & \begin{array}{ccc} x_1 & x_5 & x_6 \end{array} \\ C_e^T = & [\begin{array}{ccc} 0 & 10 & 10 \end{array}] \end{array}$$

On a trouvé *un* optimum, mais x_1 est a zéro. Ceci indique une solution non unique, car on peut faire entrer x_1 dans la base à coût constant.

Quatrième itération – suite et fin

Donc si on fait de nouveau entrer x_1 (le min) entre dans la base.

- $P = B^{-1} A_3 = [\ -2 \quad -2 \quad 1.25 \quad 1\ ]^T$
- ratios :

$$\begin{array}{ccccc} & x_4 & x_3 & x_2 & x_7 \\ \frac{\bar{b}}{P} = & 17 & 7 & 2 & -4.25 \end{array}$$

donc x_2 (le min dont P est positif) sort de la base. On a découvert un cycle.

Solution

- La solution est constituée des variables de base réalisables possibles. Ici
$$VB_1 = \{x_4, x_3, x_1, x_7\} VB_2 = \{x_4, x_3, x_2, x_7\}$$
et de toutes leurs combinaisons linéaires intermédiaires.
- Les valeurs de ces variables sont données par
$$\bar{b}_1 = \{24, 8, 2, 5\} \bar{b}_2 = \{27.2, 11.2, 1.6, 3.4\}$$
respectivement. Toutes les autres valeurs sont à 0.
- En ne comptant que les variables entrant dans le coût, les deux points optimaux extrêmes sont :

$$e_1 = \begin{bmatrix} x_1 = & 2 \\ x_2 = & 0 \\ x_3 = & 8 \end{bmatrix}, e_2 = \begin{bmatrix} x_1 = & 0 \\ x_2 = & 1.6 \\ x_3 = & 11.2 \end{bmatrix}$$

Solutions

- Toutes les solutions intermédiaires sont données par :
$$\begin{bmatrix} x_1 = & 2c \\ x_2 = & 1.6 - 1.6c \\ x_3 = & 11.2 - 3.2c \end{bmatrix}$$
avec $0 \leq c \leq 1$.
- La fonction de coût vaut donc :
$$z = -60x_1 - 30x_2 - 20x_3 = -280$$
et elle est constante pour toutes ces solutions.
- Pour des problèmes plus complexes, on peut avoir plusieurs variables à zéro dans P, et non plus une seule. L'ensemble des solution est alors l'espace vectoriel convexe induit par les solutions extrêmes. Pour les trouver il faut réaliser toute les substitutions possibles autorisées.

Solutions non bornées

- Soit une boulangerie qui fabrique des petits pains ordinaires et des pains campagnards ;
- Les pains ordinaires se vendent pour 36 centimes et les pains campagnards 30 centimes ;
- Un pain ordinaire nécessite une dose de levure et 60g de farine, un pain campagnard une dose de levure et 50g de farine.
- La boulangerie possède pour le moment 5 doses de levure et 100g de farine.
- La levure coûte 3 centime la dose, et la farine 4 centimes les 10g.
- Maximisez le profit de la boulangerie.

Cas limites de la programmation linéaire
PL dégénérée et convergence

Formulation

- x_1 = nombre de pains ordinaires produits
- x_2 = nombre de pains campagnards produits
- x_3 = nombre de doses de levure
- x_4 = farine consommée par quantité de 10g.
- Revenus = $36x_1 + 30x_2$, coûts = $3x_3 + 4x_4$.
- Objectif = $\max z = 36x_1 + 30x_2 - 3x3 - 4x_4$
- Contraintes 1 : $x_1 + x_2 \leq 5 + x_3$
- Contraintes 2 : $6x_1 + 5x_2 \leq 10 + x_4$

Forme standard

- On introduit 2 variables d'écart pour les contraintes, x_5, x_6 qui sont toutes deux positives.
- La forme standard est la suivante :

$$\begin{array}{lrrrrrrrrrrrrl}
\min & z = & -36x_1 & - & 30x_2 & + & 3x_3 & + & 4x_4 & & & & & \\
 & & x_1 & + & x_2 & - & x_3 & & & + & x_5 & & & = 5 \\
 & & 6x_1 & + & 5x_2 & & & - & x_4 & & & + & x_6 & = 10
\end{array}$$

Première itération

- VB=$\{x_5, x_6\}$; VHB=$\{x_1, x_2, x_3, x_4\}$
- $\bar{b} = \left[\begin{array}{cc} 5 & 10 \end{array}\right]$
- Coûts réduits = $\left[\begin{array}{cccc} -36 & -30 & 3 & 4 \end{array}\right]$ on va donc faire rentrer x_1.
- $P = \left[\begin{array}{cc} 1 & 6 \end{array}\right]$
- Ratios = $\left[\begin{array}{cc} 5 & 1.66667 \end{array}\right]$ on va donc faire sortir x_6.

Deuxième itération

- VB=$\{x_5, x_1\}$; VHB=$\{x_2, x_3, x_4, x_6\}$
- $\bar{b} = \begin{bmatrix} 3.333 & 1.667 \end{bmatrix}$
- Coûts réduits = $\begin{bmatrix} 0 & 3 & -2 & 6 \end{bmatrix}$ on va donc faire rentrer x_4.
- $P = \begin{bmatrix} 0.167 & -0.167 \end{bmatrix}$
- Ratios = $\begin{bmatrix} 20 & -10 \end{bmatrix}$ on va donc faire sortir x_5.

Troisième itération

- VB=$\{x_4, x_1\}$; VHB=$\{x_2, x_3, x_5, x_6\}$
- $\bar{b} = \begin{bmatrix} 20 & 5 \end{bmatrix}$
- Coûts réduits = $\begin{bmatrix} 2 & -9 & 12 & 4 \end{bmatrix}$ on va donc faire rentrer x_3.
- $P = \begin{bmatrix} 0.167 & -0.167 \end{bmatrix}$
- Ratios = $\begin{bmatrix} -6 & -1 \end{bmatrix}$ La solution n'est pas bornée.

Solution pour la boulangerie

- Avec la dernière solution de base réalisable, le système s'écrit :

$$\begin{array}{lrl} x_4 & -6x_3 & = 20 \\ x_1 & -x_3 & = 5 \end{array}$$

Donc en augmentant x_3 de façon arbitraire, on fait aussi croître x_4 et x_1 arbitrairement, et donc on peut réduire z sans fin, tout en obéissant à toutes les contraintes.

Exemple d'un problème dégénéré

- Soit le problème suivant

$$\begin{array}{lrrrrrrrrl} \min & z = & -5x_1 & - & 2x_2 & & & & & \\ & & x_1 & + & x_2 & + & x_3 & & & = 6 \\ & & x_1 & - & x_2 & & & + & x_4 & = 0 \end{array}$$

- Le simplexe ne trouve pas l'optimum !
- Il faut détecter les cycles, ce qui n'est pas toujours facile
- Heureusement ce type de problème est rare en pratique.

Programmation linéaire – 4e partie

Base de départ – dualité – application aux graphes

Hugues Talbot

Laboratoire A2SI

8 avril 2009

Plan

Trouver une base de départ

- Soit une fabrique de jus d'orange, qui vent une boisson composée de soda et de jus d'orange
- Chaque dl de soda contient 0.05kg de sucre et 1g de vitamines C
- Chaque dl de jus d'orange contient 0.025kg de sucre et 3g de vitamine C
- Chaque dl de soda coûte 2 centimes et chaque dl de jus d'orange coûte 3 centimes
- Chaque bouteille d'un litre de produit final doit contenir au moins 20g de vitamine C et au plus 0.4kg de sucre.
- Produire le tout au coût *minimal*.

Forme standard

- avec x_1 = nb de dl de soda, x_2 = nb de dl de J.O.
-

$$\begin{array}{llrcrcrcrl} \min & z = & 2x_1 & + & 3x_2 & & & & & \\ & & 0.5x_1 & + & 0.25x_2 & + & x_3 & & & = 4 \\ & & x_1 & + & 3x_2 & & & - & x_4 & = 20 \\ & & x_1 & + & x_2 & & & & & = 10 \end{array}$$

Toutes les variables positives.

- Comment trouver une SBR initiale ?

Ajout de variables

- On pourrait ajouter d'autres variables artificielles positives, par exemple :

$$\begin{array}{llrcrcrcrcrcrcl}
\min & z = & 2x_1 & + & 3x_2 & & & & & & & & & & \\
 & & 0.5x_1 & + & 0.25x_2 & + & x_3 & & & & & & & & = 4 \\
 & & x_1 & + & 3x_2 & & & - & x_4 & + & x_5 & & & & = 20 \\
 & & x_1 & + & x_2 & & & & & & & + & x_6 & & = 10
\end{array}$$

- **NOTE** Il n'est pas suffisant de rajouter une seule variable en ligne 4. En effet il n'y a pas moyen de rendre x_4 positive dans ce cas.
- Mais le système peut converger vers une solution qui n'est pas celle du pb de départ, par exemple x_3 =4, x_4=20, x_5=10, $x_1 = x_2 = 0$.

Variables artificielles

- Pour éviter ça on peut faire entrer les variable supplémentaires dans le coût, en leur donnant un coût arbitraire élevé, qui les fera converger vers 0, par exemple

$$\min z = 2x_1 + 3x_2 + Mx_5 + Mx_6$$

Avec M positif et suffisamment grand.

Itération 1 – ajout de variables

On prend $M = 100$.

- VB=$\{x_3, x_5, x_6\}$; VHB=$\{x_1, x_2, x_4\}$
- $\bar{b} = \begin{bmatrix} 4 & 20 & 10 \end{bmatrix}$
- Coûts réduits = $\begin{bmatrix} -198 & -397 & 100 \end{bmatrix}$ on va donc faire rentrer x_2.
- $P = \begin{bmatrix} 0.25 & 3 & 1 \end{bmatrix}$
- Ratios = $\begin{bmatrix} 16 & 6 & 10 \end{bmatrix}$ on va donc faire sortir x_5.

Itération 2 – ajout de variables

- VB=$\{x_3, x_2, x_6\}$; VHB=$\{x_1, x_4, x_5\}$
- $\bar{b} = \begin{bmatrix} 2.333 & 6.667 & 3.333 \end{bmatrix}$
- Coûts réduits = $\begin{bmatrix} -65.667 & -32.333 & 132.333 \end{bmatrix}$ on va donc faire rentrer x_1.
- $P = \begin{bmatrix} 0.417 & 0.333 & 0.667 \end{bmatrix}$
- Ratios = $\begin{bmatrix} 5.6 & 20 & 5 \end{bmatrix}$ on va donc faire sortir x_6.

Itération 3 – ajout de variables

- VB=$\{x_3, x_2, x_1\}$; VHB=$\{x_4, x_5, x_6\}$
- $\bar{b} = \begin{bmatrix} 0.25 & 5 & 5 \end{bmatrix}$
- Coûts réduits = $\begin{bmatrix} 0.5 & 99.5 & 98.5 \end{bmatrix}$
- Solution optimale =

$$\begin{bmatrix} x_3 = & 0.25 \\ x_2 = & 5 \\ x_1 = & 5 \end{bmatrix}$$

$z = 25.$

Solutions impossibles

- Dans le cas où on ajoute des variables artificielles, on peut tomber dans le cas où elles sont présentes dans la solution optimale, même avec M élevé. Cela indique une solution infaisable.
- Exemple : exigeons qu'une bouteille de produit final contienne 36g de vitamine C. Puisque le maximum faisable est 30g (avec de l'orange pure), on a clairement un problème.
- En faisant tourner l'algorithme, on trouve une solution en 2 itérations :

$$\begin{bmatrix} x_3 = & 1.5 \\ x_5 = & 6 \\ x_2 = & 10 \end{bmatrix}$$

$z = 630.$

Comme x_5 est artificielle, la solution n'est pas valable.

Solution en deux phases

- On choisit M en général $100\times$ plus grand que le plus grand coefficient dans la fonction objectif. Ce qui peut introduire des erreurs d'arrondis.
- Pour cette raison on préfère une solution en deux phases :
- Dans la première phase, comme précédemment on ajoute des variables artificielles, de façon à trouver une base initiale triviale au PL modifié.
- On modifie aussi la fonction objectif : on cherche maintenant à minimiser la somme des variables artificielles ajoutées. Ce cette manière on va vite converger vers une solution telle que ces variables artificielles soient nulles.

Phase II

- On a alors 3 cas :
  1. La valeur optimale (somme des variables artificielles) est non-nulle : le problème initial n'a pas de solution réalisable.
  2. La valeur optimale est nulle, et aucune variable artificielle se trouve dans la base finale : dans ce cas on prend la base finale comme base initiale du PL de départ, et on résout ce problème normalement (sans les variables artificielles).
  3. La valeur optimale est nulle, mais au moins une variable artificielle se trouve dans la base finale : dans ce cas on est obligé de garder les variables artificielles non-nulles pour résoudre le problème initial, en partant de la base obtenue comme base initiale.

Éléments de justification

- Si le PL originel n'est pas réalisable, alors la seule façon d'obtenir une solution réalisable dans le PL modifié serait d'avoir au moins une variable artificielle positive. Dans ce cas la somme des variables artificielles est non-nulle.
- D'un autre côté, si le PL originel possède une solution réalisable, alors cette solution est également réalisable dans le PL modifié, avec toutes les variables artificielles à zéro.

Itération 1 – première phase

Exemple des boissons :

- VB=$\{x_3, x_5, x_6\}$; VHB=$\{x_1, x_2, x_4\}$
- $\bar{b} = \begin{bmatrix} 4 & 20 & 10 \end{bmatrix}$
- Coûts réduits = $\begin{bmatrix} -2 & -4 & 1 \end{bmatrix}$ on va donc faire rentrer x_2.
- $P = \begin{bmatrix} 0.25 & 3 & 1 \end{bmatrix}$
- Ratios = $\begin{bmatrix} 16 & 6.667 & 10 \end{bmatrix}$ on va donc faire sortir x_5.

Itération 2 – ajout de variables

- VB=$\{x_3, x_2, x_6\}$; VHB=$\{x_1, x_4, x_5\}$
- $\bar{b} = \begin{bmatrix} 2.333 & 6.667 & 3.333 \end{bmatrix}$
- Coûts réduits = $\begin{bmatrix} -0.667 & -0.333 & 1.333 \end{bmatrix}$ on va donc faire rentrer x_1.
- $P = \begin{bmatrix} 0.417 & 0.333 & 0.667 \end{bmatrix}$
- Ratios = $\begin{bmatrix} 5.6 & 20 & 5 \end{bmatrix}$ on va donc faire sortir x_6.

Itération 3 – ajout de variables

- VB=$\{x_3, x_2, x_1\}$; VHB=$\{x_4, x_5, x_6\}$
- $\bar{b} = \begin{bmatrix} 0.25 & 5 & 5 \end{bmatrix}$
- Coûts réduits = $\begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$
- Solution optimale =

$$\begin{bmatrix} x_3 = & 0.25 \\ x_2 = & 5 \\ x_1 = & 5 \end{bmatrix}$$

$z' = 0$.

Phase II

- La solution pour la phase I est une SBR pour le problème initial.
- On utilise la base obtenue comme point de départ.
- Ici, par chance la base initiale est également la solution optimale du problème initial, mais ce n'est pas nécessairement le cas bien sûr.
- Dans le cas où on a un problème infaisable (par exemple 36g de vitamine C comme contrainte, on peut vérifier qu'on trouve comme solution :

$$\begin{bmatrix} x_3 = & 1.5 \\ x_5 = & 6 \\ x_2 = & 10 \end{bmatrix}$$

$z' = 6$
Note : tout comme pour la méthode avec M.

Variables non nécessairement positives (nnp)

- Le test des ratios impose de ne regarder que les ratios positifs, ce qui ne marche pas si certaines variables peuvent être légitimement négatives.
- Comme dans le cas précédent, on va se réduire au cas positif en ajoutant des variables.
- Pour toute variable x_i non nécessairement positive (nnp), on substitue cette variable par $x_i' - x_i''$ et on ajoute les contraintes $x_i' \geq 0$ et $x_i'' \geq 0$.
- On se retrouve donc bien dans le cadre du simplexe.
- Comme on va le voir, aucune SBR ne peut avoir à la fois $x_i' > 0$ et $x_i'' > 0$.

Variables nnp – suite

- On a alors 3 cas :
  1. $x_i' > 0$ et $x_i'' = 0$. Dans ce cas $x_i = x_i'$.
  2. $x_i' = 0$ et $x_i'' > 0$. Dans ce cas $x_i = -x_i''$.
  3. $x_i' = 0$ et $x_i'' = 0$. Dans ce cas $x_i = 0$.

Choix d'une base de départ | Cas avec des variables pouvant être négatives | Dualité | Réseaux - flots maximaux | Non-abordés

Boulangerie II

- Un boulanger possède 30kg de farine et 5 paquets de levure
- Une fournée nécessite 5kg de farine et un paquet de levure
- Chaque fournée se vend pour 30 Euros
- Le boulanger peut acheter ou revendre de la farine à 4 Euros/kg.
- Maximiser le profit.

Formulation

- x_1 = nombre de fournées ; x_2 = nombre de kg de farine supplémentaires.
- Si $x_2 > 0$ alors le boulanger a acheté de la farine, si $x_2 < 0$ il en a revendu.
- Le PL est :

$$\begin{array}{rcrcrl} \max z & = & 30x_1 & - & 4x_2 & \\ & & 5x_1 & - & x_2 & \leq 30 \\ & & x_1 & & & \leq 5 \end{array}$$

$x_1 \geq 0$, x_2 nnp.

Re-formulation

- On remplace x_2 par $x_2' - x_2''$.
- Sous forme standard :

$$\begin{array}{rcrcrcrcrcrl} \max z & = & -30x_1 & + & 4x_2' & - & 4x_2'' & & & & & \\ & & 5x_1 & - & x_2 & + & x_2'' & + & x_3 & & & = 30 \\ & & x_1 & & & & & & & + & x_4 & = 5 \end{array}$$

avec $x_1, x_2', x_2'', x_3, x_4 \geq 0$

Solution

- En 3 itérations, on trouve la solution suivante :

$$\begin{bmatrix} x_1 = & 5 \\ x_2'' = & 5 \end{bmatrix}$$

  $z = -170$
- Soit $x_2 = -5$.
- Interprétation : le boulanger doit revendre 5kg de farine. Il est optimal de faire ainsi, car les 5 paquets de levure limitent la production, et il restent $30 - 25 = 5$kg de farine à la fin, qu'il vaut mieux revendre...

Consistance des variables

- Les variables x_i' et x_i'' ne peuvent pas être simultanément positives.
- En effet les colonnes liées à x_i' et x_i'' forment deux vecteurs opposés, il ne sont pas indépendants et ne peuvent faire partie simultanément d'une base.
- Donc soit l'un d'entre eux fait partie des VB, auquel cas l'autre est nul, ou bien les deux sont nuls.

Problème primal

- Soit un PL *primal* sous forme standard :

$$
\begin{array}{rcccccccl}
\max z = & c_1x_1 & + & c_2x_2 & + & \dots & + & c_nx_n & \\
& a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & \leq b_1 \\
& a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & \leq b_2 \\
& \vdots & + & \vdots & + & & + & \vdots & \\
& a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & \leq b_m
\end{array}
$$

$\forall i, x_i \geq 0$

- le dual de ce problème devient un problème min si le primal est max, et vice-versa.

Problème dual

- Voici le problème *dual* sous forme standard :

$$
\begin{array}{rcccccccl}
\min w = & b_1y_1 & + & b_2y_2 & + & \dots & + & b_my_n & \\
& a_{11}y_1 & + & a_{21}y_2 & + & \dots & + & a_{m1}y_n & \geq c_1 \\
& a_{12}x_1 & + & a_{22}x_2 & + & \dots & + & a_{m2}x_n & \geq c_2 \\
& \vdots & + & \vdots & + & & + & \vdots & \\
& a_{1n}x_1 & + & a_{2n}x_2 & + & \dots & + & a_{mn}x_n & \geq c_n
\end{array}
$$

$\forall i, y_i \geq 0$

- le dual de ce problème devient un problème min si le primal est max, et vice-versa.

Exemple – vitamines

- Soit une personne doit absorber suffisamment de deux vitamines V et W chaque jour.
- Ces vitamines sont présentes dans des nourritures différentes, par exemple le lait et les oeufs.
- Les besoins sont résumés dans la table suivante :

Vitamine	Quantité dans le lait	Quantité dans les oeufs	Besoin journaliers
V	2	4	40
W	3	2	50
Coût unitaire	3	2.5	

- Mimiser le coût du régime.

Formulation

x_1 = quantité de lait achetée ; x_2 = quantité d'oeufs.

$$\begin{array}{rcccccccl}
\min z = & 3x_1 & + & 2.5x_2 & & & & & \\
 & 2x_1 & + & 4x_2 & - & x_3 & & & = 40 \\
 & 3x_1 & + & 2x_2 & & & - & x_4 & = 50
\end{array}$$

$x_i \leq 0.$

Solution

- On trouve, en une itération en partant de la base VB=$\{x_1, x_2\}$

$$\begin{bmatrix} x_1 = & 15 \\ x_2 = & 2.5 \end{bmatrix}$$

$z = 51.25$.

- Cette formulation est du point de vue de l'acheteur, voyons ce que donne le problème dual :

Formulation duale

y_1 = prix de l'unité de vitamine V ; y_2 = prix de l'unité de vitamine W.

$$\begin{array}{llllllll} \max w = & 40y_1 & + & 50y_2 & & & & \\ & 2y_1 & + & 3y_2 & + & y_3 & & = 3 \\ & 4y_1 & + & 2y_2 & & & + \quad y_4 & = 2.5 \end{array}$$

$y_i \geq 0$

Solution

- En partant de la base VB=$\{y_3, y_4\}$, on trouve en 3 itérations

$$\begin{bmatrix} y_1 = & 0.1875 \\ y_2 = & 0.8750 \end{bmatrix}$$

$w = 51.25$

- Cette solution donne une perspective du point de vue du vendeur, qui cherche à vendre des vitamines V et W au meilleur prix possible.

Choix d'une base de départ · Cas avec des variables pouvant être négatives · Dualité · Réseaux - flots maximaux · Non-abordés

Intérêt

- Interpretation
- Peut être plus facile à résoudre, dans le cas où le primal a beaucoup de contraintes.
- Si le primal a une matrice B de rang m, le dual a une matrice de rang n. L'une des deux formulations est souvent avantageuse en complexité de calcul.

Exemple

- Soit une entreprise de production d'électricité, possédant 3 centrales et devant alimenter 4 villes, suivant la table des coûts d'acheminement suivants :

	Ville 1	Ville 2	Ville 3	Ville 4	Prod. (GWH)
Centrale 1	8	6	10	9	35
Centrale 2	9	12	13	7	50
Centrale 3	14	9	16	5	40
Demande (GWH)	45	20	30	30	

- Formulez un PL minimisant le coût d'acheminement permettant de répondre à la demande de chaque ville.

Formulation

- On note x_{ij} le nombre de GWH produits à la centrale i et consommé à la ville j.
- Coût total = $\sum a_{ij}x_{ij}$
- Contraintes de production : exemple

$$x_{11} + x_{12} + x_{13} + x_{14} \leq 35$$

- Contraintes de consommation : exemple

$$x_{11} + x_{21} + x_{31} \geq 45$$

Problèmes non abordés en PL :

- Analyse de la robustesse de la solution.
- ...

Programmation linéaire en nombres entiers

Formulation

Hugues Talbot

Laboratoire A2SI

20 mars 2009

Plan

Nombres entiers vs. nombres réels

- Jusqu'à présent, nous avons vu des problèmes de PL avec contraintes et des variables en nombres réels (positifs pour le simplexe)
- Une manière d'étendre le problème est d'exiger qu'un ou l'autre de ces aspects du problème s'exprime en nombres entiers.
- Dans le cas où les contraintes *et* les variables sont toutes deux entières, on parle de programmation linéaire en nombre entiers (ou simplement programmation en nombres entiers - IP en anglais).
- Dans le cas ou seulement un de ces aspects s'exprime en nombre entiers, ou même seulement certaines des variables, on parle de programmation linéaire mixte (MP en anglais).

Exemple

- Maximiser $z = x_1 + x_2$
- avec

$$\begin{array}{rcrcr} -2x_1 & + & 2x_2 & \geq & 1 \\ -8x_1 & + & 10x_2 & \leq & 13 \end{array}$$

- et $x_1, x_2 \geq 0$
- Avec le simplexe, en nombres réels, on trouve l'optimum $\{x_1 = 4, x_2 = 9/2\}$.
- Si le problème est contraint en nombres entiers (x_i sont des unités indivisibles), alors l'optimum est $\{x_1 = 1, x_2 = 2\}$!
- Comment passer de la solution continue à la solution entière ? Ici prendre l'entier le plus proche n'est pas faisable...

Catégories de problèmes en IP

1. Problèmes avec entrées/sorties discrètes : production d'objets, etc.
2. Problèmes avec conditions logiques : ajout de variables entières avec des contraintes supplémentaires. (par exemple : si le produit A est fabriqué alors produire également B ou C)...
3. Problèmes combinatoires : séquençage, allocation de ressources, emplois du temps, TSP : formulables en IP.
4. Problèmes non linéaires : souvent formulables en IP. C'est utile en particulier quand la région réalisables est non-convexe.
5. Problèmes de réseaux, problèmes de graphes – exemple : colorier une carte.

Remarques sur la formulation en IP

- La plupart des modèles sont dans la seconde catégorie (pb avec conditions logiques), la plupart des problèmes sont donc des problèmes mixtes avec quelques variables entières ajoutées ;
- IP peut être un moyen (avec de l'expérience) pour modéliser effectivement une grande classe de problèmes : contrairement à ce qu'on pourrait penser, contraindre davantage le problème (IP vs. LP) permet de modéliser plus de problèmes !
- Cependant, modéliser n'est pas résoudre ! On verra que IP est en général NP-complet, et donc les méthodes de résolutions sont non-linéaire en complexité...
- La complexité de la résolution dépend de la formulation !

Variables de décision

- Parmi les variables entières, il existe la classe des variables booléennes, pouvant prendre comme valeur 0 ou 1.
- Ces variables sont souvent utilisées en MP pour représenter des décisions : implications, liens entre variables, etc.
- Exemple des variables d'affectation et indicatrices.

Exemple : Choix

- Soient 4 choix possibles, chacun nécessitant des moyens et donnant un rendement

choix	moyen	rendement
1	5 unités	16 unités
2	7 ...	22 ...
3	4 ...	12 ...
4	3 ...	8 ...

- Moyens disponibles : 14 unités
- Maximiser le rendement.
- Extentions :
  1. On doit effectuer au plus 2 choix
  2. Choix 2 ne peut être effectué que si 1 est également pris

Affectation : formulation

- On utilise des variables booléennes (0 et 1), par exemple

$$X_j, j \in \{1, \ldots, 4\} = \begin{cases} 1 & \text{Choix effectué} \\ 0 & \text{Sinon} \end{cases}$$

- On cherche à maximiser $z = 16x_1 + 22x_2 + 12x_3 + 8x_4$
- Avec la contrainte

$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

- Contraintes optionnelles :
  1. $x_1 + x_2 + x_3 + x_4 \leq 2$
  2. $x_1 \geq x_2 \Leftrightarrow -x_1 + x_2 \leq 0$

Variables indicatrices

- On suppose que x modèle une quantité (réelle) d'un ingrédient à inclure dans un mélange. On cherche à distinguer le cas $x = 0$ du cas $x > 0$.
- On introduit la variable δ, qui prend la valeur 1 lorsque $x > 0$, avec la contrainte :

$$x - M\delta \leq 0,$$

  avec M un coefficient connu représentant une valeur maximale possible de x.
- Avec cette contrainte, on a bien $x > 0 \Rightarrow \delta = 1$.

Indicatrices : suite

- Pour l'implication opposée ($x = 0 \Rightarrow \delta = 0$), c'est plus problématique. Cette implication est équivalente à $\delta = 1 \Rightarrow x > 0$.
- On introduit une implication moins sévère : $\delta = 1 \Rightarrow x > m$, avec m un niveau minimum acceptable en dessous duquel on considère x non-utilisé (dépendant de l'application). Une contrainte équivalente est alors :

$$x - m\delta \geq 0.$$

Problèmes à charge fixe

- Exemple : unités de production d'énergie
- Cas général de production avec coût initial et coût marginal.
- La puissance générée vaut soit $P_i = 0$, soit $P_i^m \leq P_i \leq P_i^M$.
- avec un coût :

$$C_i = \begin{cases} 0 & \text{Unité arrêtée} \\ a_i + b_i P_i & \text{Sinon} \end{cases}$$

- Produire une certaine puissance au moindre coût.
- Ici le coût est non-linéaire et même non-continu. LP n'est pas capable de modéliser le problème.
- Comment faire ?

Formulation de la charge fixe en IP

- On introduit la variable

$$x_i = \begin{cases} 0 & \text{Unité } i \text{ arrêtée} \\ 1 & \text{Sinon} \end{cases}$$

- Les contraintes

$$x_i P_i^m \leq P_i \leq x_i P_i^M$$

- avec

$$C_i = a_i x_i + b_i P_i$$

(et non $C_i = a_i x_i + b_i P_i x_i$, qui serait non-linéaire).

- Note : la condition $P_i = 0, x_i = 1$ est réalisable mais de coût supérieur au cas $P_i = 0, x_i = 0$.

Problèmes d'affectation

- On a n tâches affectée à n personnes ;
- On veut affecter une et une seule tâche à chaque personne ;
- Le rendement de l'affectation de la tâche i à la personne j est donnée par la matrice C_{ij} ;
- On veut maximiser le rendement ;
- Formulation ?
- Formulation étendue : si le nombre de tâches est inférieur au nombres de personnes.

Formulation de l'affectation

- On introduit les variables x_{ij}

$$x_{ij} = \begin{cases} 1 & \text{tâche i affectée à la personne j} \\ 0 & \text{Sinon} \end{cases}$$

- contraintes :

$$\begin{aligned} \textstyle\sum_{i=1}^{n} x_{ij} &= 1 (\text{une seule tâche affectée à } j) \\ \textstyle\sum_{j=1}^{n} x_{ij} &= 1 (\text{chaque tâche affectée une fois}) \end{aligned}$$

- Maximiser

$$\sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij}$$

Problème d'affectation étendu

- Nombre de personnes $>$ nombre de tâches, i.e n personnes, m tâches, $n > m$: $\sum_{i=1}^{m} x_{ij} \leq 1, j = 1, \ldots, n$
- Somme des tâches affectées à la personne j est ≤ 1 : $\sum_{j=1}^{n} x_{ij}, i = 1, \ldots, m$
- Somme des personnes affectées à la tâche i est $= 1$: on ajoute des variables d'écart : $s_j + \sum_{i=1}^{m} x_{ij} = 1, j = 1, \ldots, n$
- Maximiser

$$\sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij}$$

Autre façon de formuler : pb de transport

Nous verrons comment reformuler ce problème dans le dernier cours.

- (diagramme personne-tâches 4.18).
- contraintes en entiers,
- valeurs max de x_{ij} est 1, min = 0, donc solution en nombre entiers.

Conversion algèbre booléenne en algèbre linéaire avec variables binaires

- on introduit

$$\delta_i = \begin{cases} 1 & \text{si } x_i \text{ est vrai} \\ 0 & \text{sinon} \end{cases}$$

-

$$\begin{array}{ccccc} x_i & \text{vrai} & \equiv & \delta_i = 1 \\ x_i & \text{faux} & \equiv & \delta_i = 0 \end{array}$$

Opérations logiques

$\sim$	négation
$\wedge$	ET logique
$\vee$	OU logique
$\Rightarrow$	Implication
$\Leftrightarrow$	Equivalence
$\oplus$	OU exclusif
V	VRAI
F	FAUX

Quelques opérations

$$
\begin{array}{lcr}
x_1 \vee x_2 = V & \equiv & \delta_1 + \delta_2 \geq 1 \\
x_1 \wedge x_2 = F & \equiv & \delta_1 = 1, \delta_2 = 1 \\
\sim x_1 = 1 & \equiv & \delta_1 = 0 \\
x_1 \Rightarrow x_2 & \equiv & \delta_2 \geq \delta_1 \\
x_1 \Leftrightarrow x_2 & \equiv & \delta_1 = \delta_2 \\
x_1 \oplus x_2 = V & \equiv & \delta_1 + \delta_2 = 1
\end{array}
$$

Combinaison de variables logiques et continues

- Variable logique = indicateur (e.g. ouvert/fermé , chaud/froid)
- cas $x_i = [f(x_i) \leq 0]$, soit, si $\delta_i = 1$ si $x_i = V$.
- $$\begin{array}{lll} \delta_i = 1 & \text{si} & f(x_i) \leq 0 \\ \delta_i = 0 & \text{si} & f(x_i) \geq \varepsilon \text{ (précision)} \end{array}$$
- On considère
  $$\begin{array}{lll} M & = & \max_{x_i}(f(x_i)) \\ m & = & \min_{x_i}(f(x_i)) \end{array},$$
  $M > 0, m < 0$.

Equations algébriques

$$\begin{array}{lll} f(x_i) & \leq & M(1-\delta_i) \\ f(x_i) & \geq & \varepsilon + (m-\varepsilon)\delta_i \end{array}$$

1. soit $\delta_i = 1$, alors $f(x_i) \leq 0, f(x_i) \geq \varepsilon$.
2. soit $\delta_i = 0$, alors $f(x_i) \leq M, f(x_i) \geq \varepsilon$.
3. soit $f(x_i) \geq 0$, alors $\delta_i = 0$ (démo tableau)
4. soit $f(x_i) \leq 0$, alors $\delta_i = 1$ (idem)

Produit de variables binaires

- soit le produit de variables $\delta_3 = \delta_1\delta_2$: contraintes non-linéaires.
- Pour transformer en série de contraintes linéaires, on pose :

$$\begin{aligned} \delta_3 &\leq \delta_1, & \delta_3 &\leq \delta_2 \\ \delta_3 &\geq \delta_1 + \delta_2 - 1 \end{aligned}$$

On vérifie l'équivalence :

1. $\delta_1 = 0, \delta_2 = 0 \Rightarrow \delta_3 \leq 0 \Rightarrow \delta_3 = 0$
2. $\delta_1 = 1, \delta_2 = 0 \Rightarrow \delta_3 \leq 0, \delta_3 \geq 0 \Rightarrow \delta_3 = 0$
3. de même pour $\delta_1 = 0, \delta_2 = 1$
4. $\delta_1 = 1, \delta_2 = 1 \Rightarrow \delta_3 \leq 1, \delta_3 \geq 1 \Rightarrow \delta_3 = 1$
5. Inversement : $\delta_3 = 1 \Rightarrow \delta_1 \geq 1, \delta_2 \geq 1, \delta_1 + \delta_2 \leq 2$, donc $\delta_1 = \delta_2 = 1$.
6. $\delta_3 = 0 \Rightarrow \delta_1 \geq= 0, \delta_2 \geq 0, \delta_1 + \delta_2 \leq 1$ donc seulement δ_1 ou δ_2 vaut 1 (l'autre vaut 0).

Produit d'une variable binaire et d'une variable continue

- soit le produit non-linéaire de deux variables, l'une continue, l'autre booléenne : $\delta f(x)$, i.e :

$$y = \begin{cases} 0 & \text{si } \delta = 0 \\ f(x) & \text{si } \delta = 1 \end{cases}$$

- On impose les contraintes :

$$\begin{cases} y & \leq M\delta \\ y & \geq m\delta \\ y & \leq f(x) - m(1-\delta) \\ y & \geq f(x) - M(1-\delta) \end{cases}$$

Produit booléen-continu, suite

1. soit $\delta = 0$, alors $y \leq 0, y \geq 0$, donc $y = 0$.
   Egalement $f(x) - M \leq y \leq f(x) - m$, donc $f(x) - M \leq 0$ et $f(x) - m \geq 0$, ce qui est non-critique.
2. soit $\delta = 1$, alors

$$\left.\begin{array}{ll} y & \leq M \text{ (non-critique)} \\ y & \geq m \text{ (non-critique)} \\ y & \leq f(x) \\ y & \geq f(x) \end{array}\right\} y = f(x)$$

3. L'inverse est également correct : $y = f(x) \Leftrightarrow \delta = 1$.

Applications

- Systèmes dynamiques et logiques
  Exemple :

$$X(t+1) = \begin{cases} 0.8X(t) & +U(t) \quad \text{si } X(t) \geq 0 \\ -0.8X(t) & +U(t) \quad \text{si } X(t) < 0 \end{cases}$$

  avec $-10 \leq X \leq 10$, $-1 \leq U \leq 1$.
- Systèmes invariants linéaires par morceaux
- Machines à états finis.

Quelques problèmes en optimisation combinatoire

1. P.L. en nombres entiers :

$$\begin{array}{ll} \min & \mathbf{C}^T\mathbf{x} \\ & \mathbf{Ax} \quad \leq \mathbf{b} \\ & x_i \text{ entiers} \end{array}$$

2. Cas particulier : variables booléennes

$$\begin{array}{ll} \min & \mathbf{C}^T\mathbf{x} \\ & \mathbf{Ax} \quad \leq \mathbf{b} \\ & x_i = \ 0 \text{ ou } 1, \mathbf{x} \in \mathcal{B} \end{array}$$

Problèmes-types - I

3. Sac à dos (knapsack) : une seule contrainte

$$\begin{array}{ll} \max & \mathbf{C}^T\mathbf{x} \\ & \mathbf{a}^T\mathbf{x} \quad \leq b \\ & \mathbf{x} \in \mathcal{B} \end{array}$$

   Note : $\mathbf{a} > 0$, sinon, $x_i' = 1 - x_i$.

4. Sacs à dos multiples

$$\begin{array}{ll} \max & \mathbf{C}^T\mathbf{x} \\ & \mathbf{AX} \quad \leq \mathbf{b} \\ & \mathbf{x} \in \mathcal{B} \end{array}$$

   Note : $\forall i, j, A_{ij} \geq 0$.

Problèmes-types - II

5. Couplage (set matching)

$$\begin{array}{lll} \max & \mathbf{C}^T\mathbf{x} & \\ & \mathbf{AX} & \leq 1(\text{ vecteurs de 1s }) \\ & \mathbf{X} \in \mathcal{B} & \end{array}$$

   Note : $\forall i, j, A_{ij} \in \mathcal{B}, X \in \mathcal{B}$.

6. Recouvrement (set covering)

$$\begin{array}{lll} \min & \mathbf{C}^T\mathbf{x} & \\ & \mathbf{AX} & \geq 1(\text{ vecteurs de 1s }) \\ & \mathbf{X} \in \mathcal{B} & \end{array}$$

   Note : $\forall i, j, A_{ij} \in \mathcal{B}, X \in \mathcal{B}$.

Problèmes-types - III

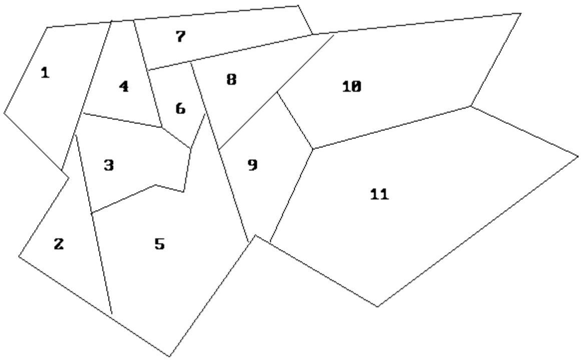
7. Partitionnement (set partitionning)

$$\begin{array}{lll} \min & \mathbf{C}^T\mathbf{x} & \\ & \mathbf{AX} & = 1(\text{ vecteurs de 1s }) \\ & \mathbf{X} \in \mathcal{B} & \end{array}$$

   Note : $\forall i, j, A_{ij} \in \mathcal{B}, X \in \mathcal{B}$.
   Cas particulier : problème d'affectation.

Exemple de problème de recouvrement

Soit la ville suivante, composée des arrondissements suivants :



On doit construire un certain nombre d'hôpitaux, qui devront s'occuper des urgences médicales provenant de l'arrondissement où il est construit et de tous ses voisins immédiats. On cherche à minimiser le nombre hôpitaux. Formulation ?

Formulation du problème des hôpitaux

- On se donne une variable x_i booléenne pour chaque arrondissement. Si x_i vaut 1, il y aura un hôpital dans l'arrondissement i, et si elle vaut 0, il n'y en aura pas.
- On formule de la façon suivante :
  min $\sum_i x_i$, avec :

$$
\begin{array}{lllllllllll}
x_1 + x_2 + x_3 + x_4 & & & & & & & \geq 1 \\
x_1 + x_2 + x_3 + x_5 & & & & & & & \geq 1 \\
x_1 + x_2 + x_3 + x_4 + x_5 + x_6 & & & & & & & \geq 1 \\
x_1 + x_3 + x_4 + x_6 + x_7 & & & & & & & \geq 1 \\
x_2 + x_3 + x_5 + x_6 + x_8 + x_9 & & & & & & & \geq 1 \\
x_3 + x_4 + x_5 + x_6 + x_7 + x_8 & & & & & & & \geq 1 \\
x_4 + x_6 + x_7 + x_8 & & & & & & & \geq 1 \\
x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} & & & & & & & \geq 1 \\
x_5 + x_8 + x_9 + x_{10} + x_{11} & & & & & & & \geq 1 \\
x_9 + x_{10} + x_{11} & & & & & & & \geq 1
\end{array}
$$

- On demande de "recouvrir" l'ensemble de manière optimale : une solution est $x_3 = x_8 = x_9 = 1$ et le reste à 0.

Conclusion

- On peut modéliser de nombreux problèmes par la programmation linéaire en nombres entiers (IP) ou par la programmation linéaire mixte (MP).
- Problèmes logiques, d'affectation, réseau, etc.
- PI et PM sont donc des outils puissants de modélisation
- Reste à résoudre ces problèmes :
  - existence d'une solution ?
  - unicité ?
  - algorithme de calcul de la solution ?
- On verra que la plupart des problèmes de PI et PM sont des problèmes difficiles à résoudre (NP-difficile).
- Deux méthodes principales de résolution : par coupes (Gomorry) et par séparation-évaluation (Branch-and-bound).

Programmation linéaire en nombres entiers

Résolution : coupes et séparation - évaluation

Hugues Talbot

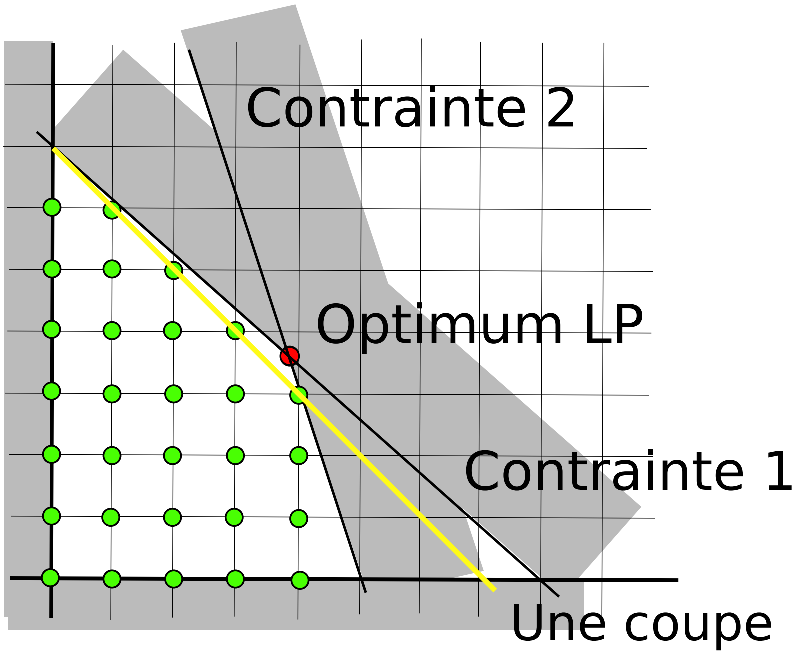
Laboratoire A2SI

26 mars 2009

Résolution des problèmes IP Solution par séparation-évaluation Algorithme général Formulation de problèmes intéressants Con

Plan

Principe des coupes



Formulation initiale

- Soit une “relaxation” linéaire d’un problème IP, c’est-à-dire le problème de PL correspondant, dont la solution (par le simplexe par exemple) optimale est X^*.
  - soit X^* est entière, auquel cas c’est la solution optimale du problème IP.
  - soit elle ne l’est pas, X^* est fractionnaire.
- Considérons la base correspondant à la solution optimale de PL
$$X_b = B^{-1}[b - EX_e]; \bar{b} = B^{-1}b.$$
avec $\bar{b}_i = B^{-1}b$ fractionnaire.
- Nous avons
$$X_{b_i} = e_i^t B^{-1}[b - EX_e]$$
où $e_i^t = [000\ldots \underset{\text{pos}\,i}{1} \ldots 00]$

Formulation (suite)

- Ainsi :

$$\begin{aligned} X_{b_i} &= e_i^t B^{-1} b - e_i^t B^{-1} E X_e \\ &= \bar{b}_i - \alpha^i X_e \end{aligned}$$

c-à-d :

$$\alpha^i = e_i^t B^{-1} E \text{ (ligne } i \text{ de } B^{-1}E)$$

- En réordonnant :

$$\begin{aligned} X_{b_i} + \alpha^i X_e &= \bar{b}_i \\ X_{b_i} + \textstyle\sum_j \alpha^i_j X_{e_j} &= \bar{b}_i \qquad (1) \end{aligned}$$

Partie entière / partie fractionnaire

- Définition : soit a un non-entier,
  - ent(a) = plus grand entier $< a$.
  - frac(a) = $a -$ ent(a)(*donc*) > 0.
- de (1) on a :

$$\begin{aligned} X_{b_i} + \textstyle\sum_j [\mathrm{ent}(\alpha^i_j) + \mathrm{frac}(\alpha^i_j)] X_{e_j} &= \mathrm{ent}(\bar{b}_i) + \mathrm{frac}(\bar{b}_i) \\ X_{b_i} + \textstyle\sum_j \mathrm{ent}(\alpha^i_j) X_{e_j} + \sum_j \mathrm{frac}(\alpha^i_j) X_{e_j} &= \mathrm{ent}(\bar{b}_i) + \mathrm{frac}(\bar{b}_i) \qquad (2) \end{aligned}$$

avec frac$(\alpha^i_j) \geq 0$ et $X_{e_j} \geq 0$.

- donc

$$X_{b_i} + \sum_j \mathrm{ent}(\alpha^i_j) X_{e_j} \leq \mathrm{ent}(\bar{b}_i) + \mathrm{frac}(\bar{b}_i)$$

Coupe de Gomory

- Mais X_{b_i} est entier et X_{e_j} aussi.
- Donc
$$X_{b_i} \sum_j \text{ent}(\alpha_j^i) X_{e_j} \leq \text{ent}(\bar{b}_i) \qquad (3)$$
- de (3)–(2) on déduit
$$\begin{array}{rcl} -\sum_j \text{frac}(\alpha_j^i) X_{e_j} & \leq & -\text{frac}(\bar{b}_i) \\ \sum_j \text{frac}(\alpha_j^i) X_{e_j} & \geq & \text{frac}(\bar{b}_i) \qquad (4) \end{array}$$

(4) est une coupe de Gomory.

Exemple

- Soit le problème suivant :
$$\begin{array}{llcll} \max & x_1 & + & x_2 & \\ & 3x_1 & + & 4x_2 & \leq 15 \\ & x_1 & - & 4x_2 & \leq 0 \end{array}$$
- Sous forme standard :
$$\begin{array}{lllll} 3x_1 & +4x_2 & +x_3 & & = 15 \\ x_1 & -4x_2 & & +x_4 & = 0 \end{array}$$

Coupe 1

- Solution optimale : base $X_b = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $B = \begin{bmatrix} 3 & 4 \\ 1 & -4 \end{bmatrix}$
- Inverse : $B^{-1} = -\frac{1}{16} \begin{bmatrix} -4 & -4 \\ -1 & 3 \end{bmatrix} = \begin{bmatrix} 1/4 & 1/4 \\ 1/16 & -3/16 \end{bmatrix}$
- Solution :
  $\bar{b} = B^{-1}b = \begin{bmatrix} 1/4 & 1/4 \\ 1/16 & -3/16 \end{bmatrix} \begin{bmatrix} 15 \\ 0 \end{bmatrix} = \begin{bmatrix} 15/4 \\ 15/16 \end{bmatrix}$
- Coupe (1) : $\alpha^1 = [\frac{1}{4}\frac{1}{4}], \bar{b}_1 = \frac{15}{4}$
  $\frac{1}{4}x_3 + \frac{1}{4}x_4 \geq \frac{3}{4}$

$$x_3 + x_4 \geq 3 \qquad \text{coupe (1)}$$

Coupe 2

- Coupe (2) :
  $\alpha^2 = [\frac{1}{16} - \frac{3}{16}], \bar{b}_2 = \frac{15}{16}$, Attention ! $\text{frac}(\alpha^2) = [\frac{1}{16}\frac{13}{16}]$
  $\frac{1}{16}x_3 + \frac{13}{16}x_4 \geq \frac{15}{16}$

$$x_3 + 13x_4 \geq 15 \qquad \text{coupe (2)}$$

- En soit pas très intéressantes : on exprime les variables hors-base en fonction des variables de base :

$$\begin{array}{rclll} x_3 & = & 15 & -3x_1 & -4x_2 \\ x_4 & = & & -x_1 & +4x_2 \end{array}$$

Coupes finales

- En substituant dans coupe 1 :

$$\begin{aligned} 15 - 3x_1 - 4x_2 - x_1 + 4x_2 &\geq 3 \\ 12 &\geq 4x_1 \\ x_1 &\leq 3 \end{aligned}$$

- En substituant dans coupe 2 :

$$\begin{aligned} 15 - 3x_1 - 4x_2 - 13x_1 + 52x_2 &\geq 15 \\ -16x_1 + 48x_2 &\geq 0 \\ x_1 - 3x_2 &\leq 0 \end{aligned}$$

- On ajoute les nouvelles coupes au problème de départ, et cette fois-ci on trouve :
$\bar{b} = [3, \frac{3}{2}]$, on doit faire une nouvelle itération.

Solution par séparation-évaluation

- On a déjà vu une méthode de solution des problèmes de IP en utilisant des coupes (de Gomory).
- Dans le cas où les contraintes et variables sont rationnelles (a fortiori entières), les coupes de Gomory convergent vers la solution optimale en temps fini.
- Les coupes attaquent le problème par l'extérieur en gardant le domaine réalisable convexe, ce qui peut être inefficace dans certaines configurations
- On peut aussi résoudre les problèmes de IP/MP en fractionnant le domaine en régions faisables bornées par des frontières alignées sur des entiers : on *sépare* ainsi les domaines, et on *évalue* quelle région explorer en premier : on appelle cette méthode la séparation-évaluation, ou *branch-and-bound* en anglais.

Les tables et les chaises

- Une entreprise fabrique des armoires et des tables ;
- Une armoire nécessite 1h de travail et 9 m^2 de bois ;
- Une table nécessite 1h de travail et 5m^2 de bois ;
- On dispose de 6h de travail et de 45 m^2 de bois ;
- Chaque armoire génère un profit de 8 €, et chaque table €;
- Formuler et résoudre en entiers.

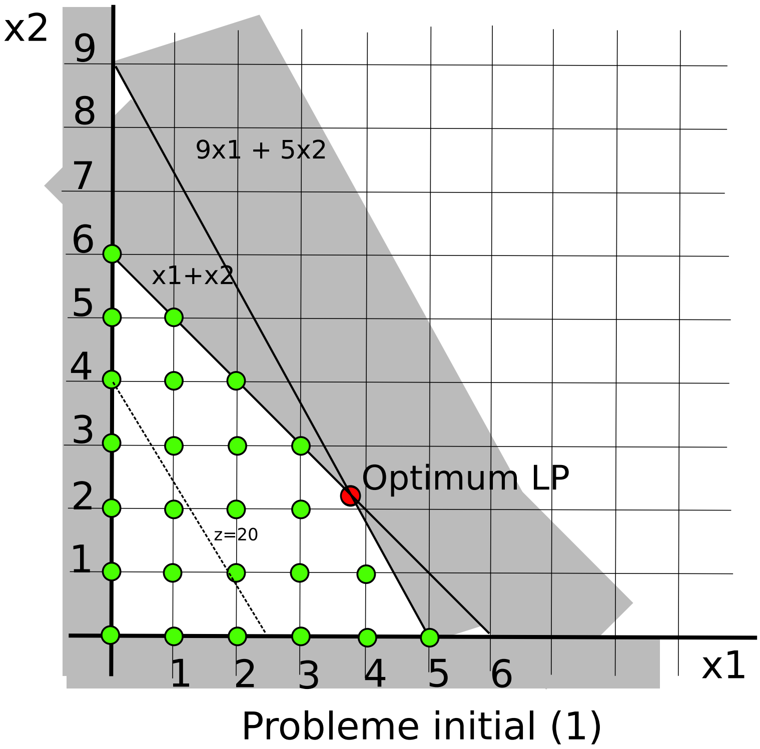
Formulation

On formule de la façon suivante :

$$\begin{array}{rcccl} \max z = & 8x_1 & + & 5x_2 & \\ & x_1 & + & x_2 & \leq 6 \\ & 9x_1 & + & 5x_2 & \leq 45 \end{array}$$

$x_i \geq 0, x_i \in \mathbb{N}.$

Représentation



Méthode générale

- On commence par résoudre la formulation de relaxation du problème en entiers, c-à-d le problème en variables continues (LP). Ici, on obtient :

$$\begin{aligned} z &= \frac{165}{4} \\ x_1 &= \frac{15}{4} \\ x_2 &= \frac{9}{4} \end{aligned}$$

- Si la solution est en entiers, on s'arrête, on a trouvé l'optimal. Ici ce n'est pas le cas, mais la valeur z obtenue est une borne supérieure pour l'optimum en entiers.

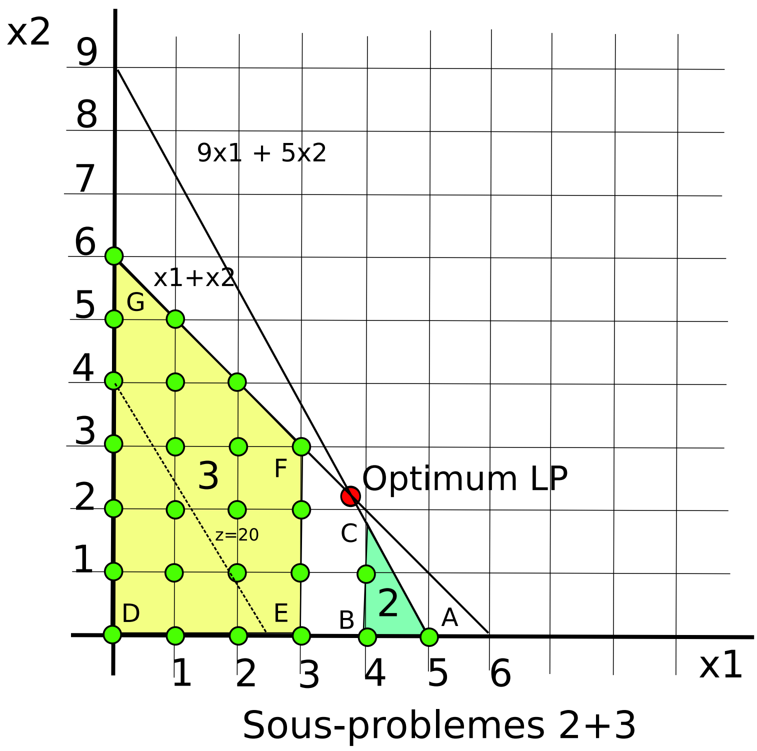
Methode générale (suite)

- Si la solution n'est pas entière, on partitionne le domaine : on choisit arbitrairement une variable qui est fractionnaire dans la solution optimale relaxée, par exemple ici x_1.
- On applique des contraintes supplémentaires dues à la nature de la variable, ici par exemple on impose soit $x_1 \leq 3$, soit $x_1 \geq 4$ (Question : pourquoi pouvons nous éliminer les solutions $3 < x_1 < 4$?)

Séparation

- On a donc maintenant deux sous-problèmes :
  2. Problème initial + contrainte $x_1 \geq 4$
  3. Problème initial + contrainte $x_2 \leq 3$.

Représentation (2)



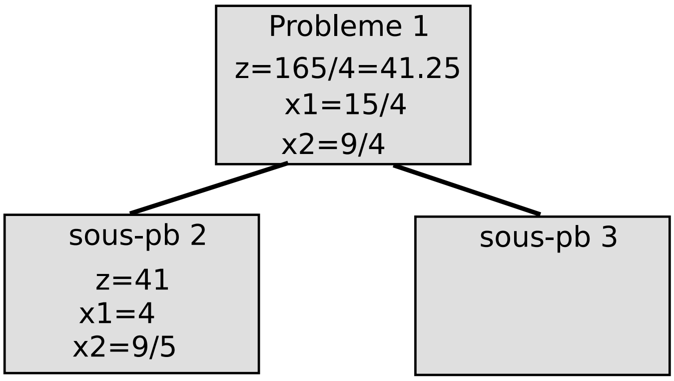
Récursion

- On constate qu'on a, ce faisant, éliminé la région contenant l'optimum de LP.
- On a maintenant deux problèmes qu'on peut de nouveau résoudre en LP.
- On en choisi un arbitrairement parmi ceux non résolus, par exemple ici le problème 2.
- La solution de relaxation (LP) pour la région 2 est

$$\begin{aligned} z &= 41 \\ x_1 &= 4 \\ x_2 &= \frac{9}{5} \end{aligned}$$

Correspondant au point C.
- On a créé la première branche d'un *arbre*.
- Comme x_2 est toujours fractionnaire, on décide de séparer sur cette variable, on sépare donc la région 2 entre deux zones : celle pour $x_2 \geq 2$ et celle pour $x_2 \leq 1$.

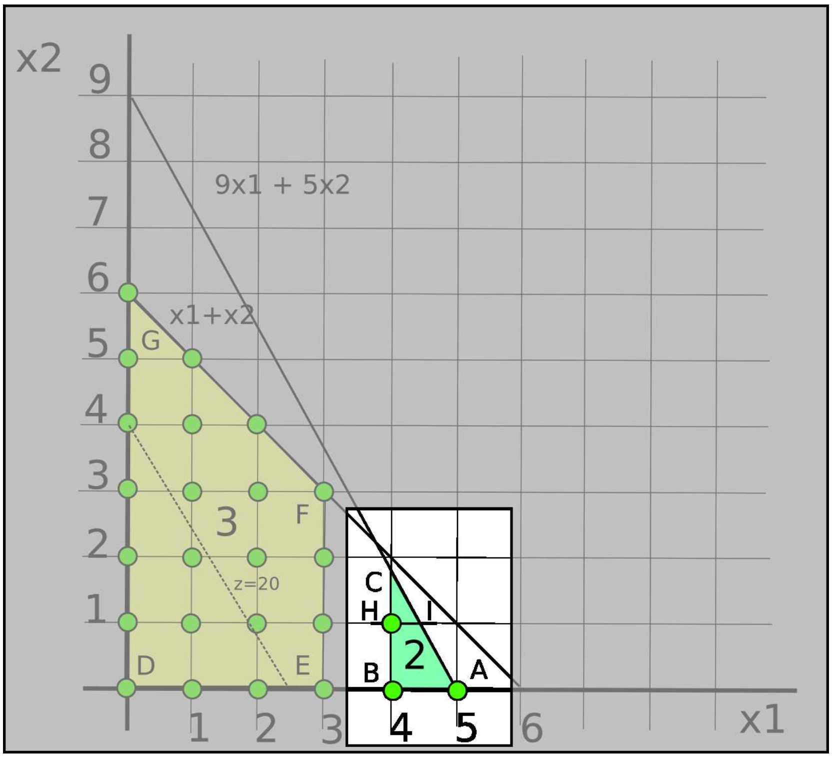
Arbre (1+2)



Nouvelle séparation (4 + 5)

- Nous voici avec deux nouveaux sous-problèmes
  4. Problème 2 + contrainte $x_2 \geq 2$
  5. Problème 2 + contrainte $x_2 \leq 1$.
- Il est préférable d'explorer l'arbre en profondeur d'abord (on verra plus tard pourquoi), on choisi donc de regarder l'un ou l'autre des nouveaux sous-problèmes, par exemple le sous-problème 4. Or ce problème n'est pas réalisable en entiers.

Représentation (4 et 5)



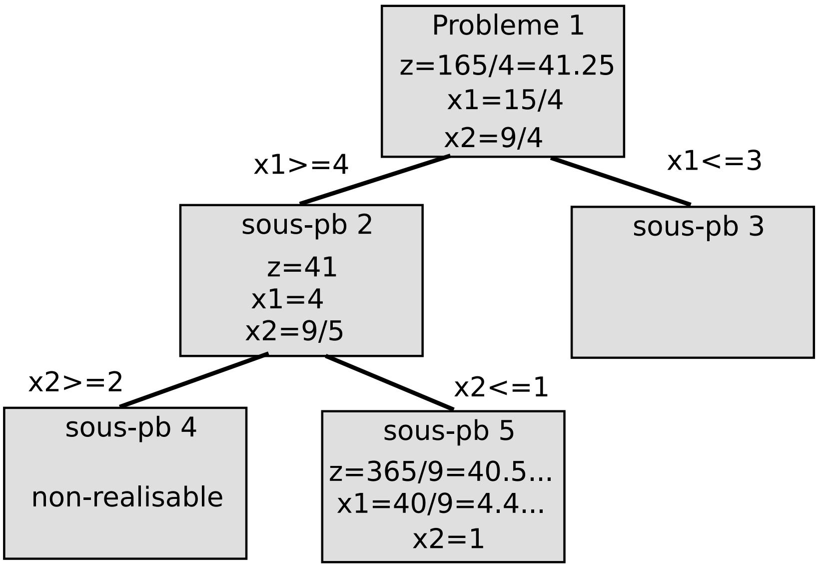
Évaluation (4 + 5)

- On constate que le problème 4 (région $x_2 \geq 2$) n'est pas réalisable.
- En résolvant le problème de relaxation LP lié au sous-problème 5, on trouve l'optimum I avec

$$\begin{aligned} z &= 365/9 = 40.555\ldots \\ x_1 &= \tfrac{40}{9} = 4.444\ldots \\ x_2 &= 1 \end{aligned}$$

- il faut donc de nouveau séparer sur x_1, avec les contraintes
  6. Problème 5 + contrainte $x_1 \geq 5$
  7. Problème 5 + contrainte $x_2 \leq 4$.

Arbre (4+5)



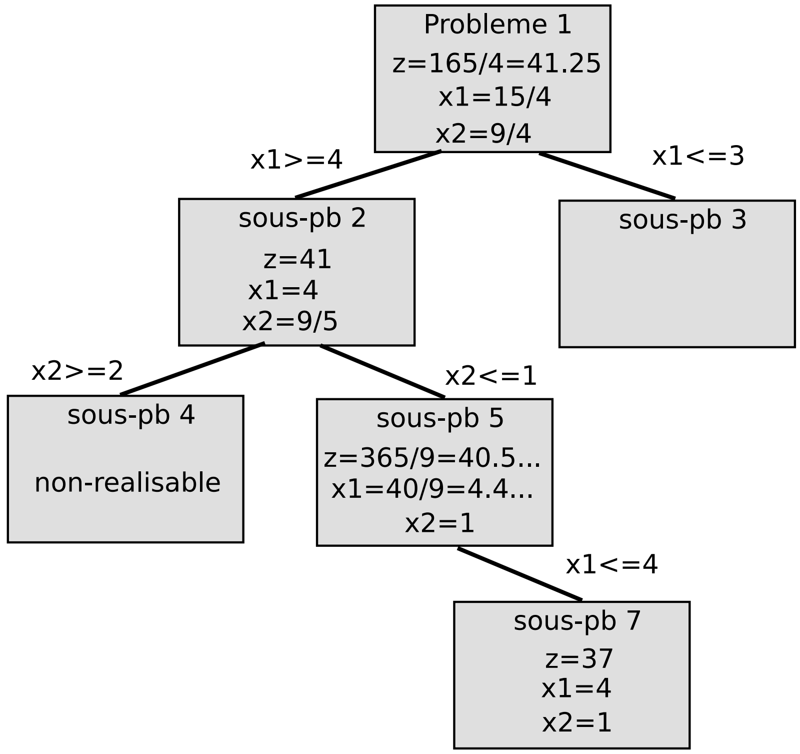
Séparation (6+7)

- Nous voici avec deux nouveaux problèmes, comme d’habitude, on va éliminer la solution fractionnaire en séparant de part et d’autre.
- Nous procédons en profondeur d’abord parmi les nouveaux sous-problèmes non-résolus (6 et 7). Nous choisissons arbitrairement 7.
- La solution de relaxation LP est maintenant :

$$\begin{aligned} z &= 37 \\ x_1 &= 4 \\ x_2 &= 1 \end{aligned}$$

  Cette solution est réalisable et correspond au point H, c’est une *solution candidate* qui nous donne une *borne inférieure* sur notre résultat final.
- Il est inutile de continuer à séparer sur cette branche.

Arbre (7)



Résolution des problèmes IP Solution par séparation-évaluation Algorithme général Formulation de problèmes intéressants Con

Évaluation (6)

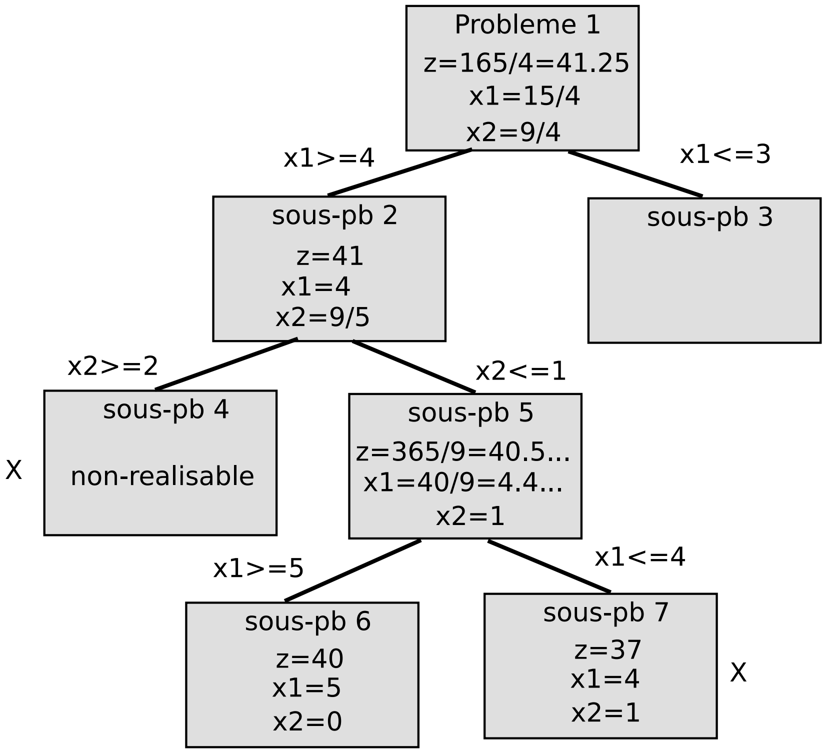
- On continue à évaluer en profondeur d'abord, soit maintenant le sous-problème 6.
- On trouve la solution correspondant au point A :

$$\begin{aligned} z &= 40 \\ x_1 &= 5 \\ x_2 &= 0 \end{aligned}$$

Il s'agit également d'une solution réalisable candidate. La valeur de la borne inférieure de notre problème est donc maintenant 40.

- La solution 7 n'est donc pas optimale.
- Il est inutile de séparer davantage sur 6.

Arbre (6)



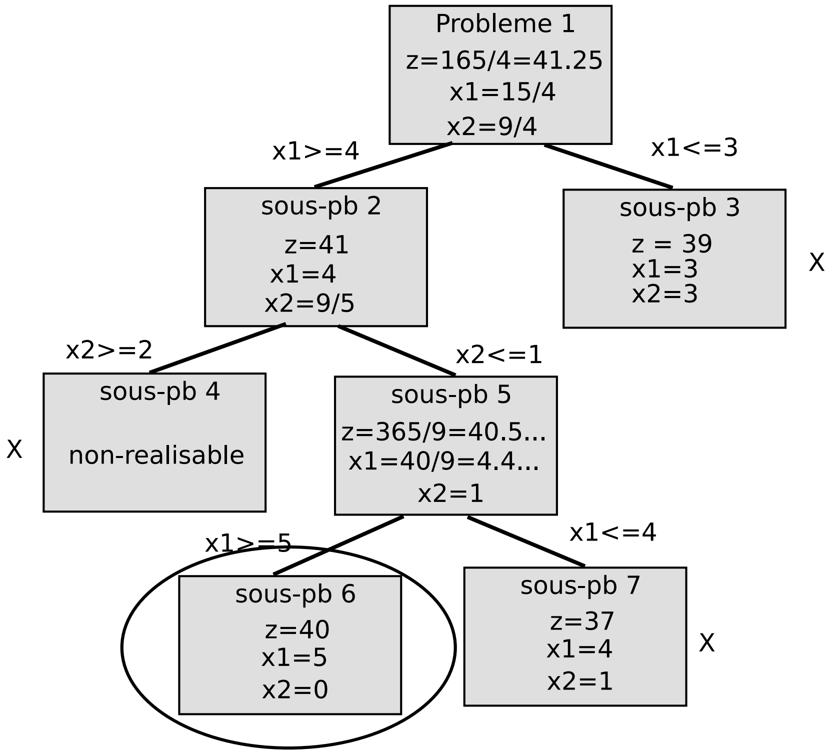
Évaluation (3)

- Il reste à évaluer la solution pour (3). On trouve la solution correspondant au point F :

$$\begin{aligned} x_1 &= 3 \\ x_2 &= 3 \\ z &= 39 \end{aligned}$$

- Ce résultat est inférieur à 40, notre borne inf, donc cette branche de l'arbre de peut pas produire un meilleur résultat que celui déjà connu correspondant au point A.
- Il ne reste plus de nœud de l'arbre à explorer, on a donc trouvé notre optimum en nombre entier : fabriquer 5 armoires et 0 table pour 40€.

Arbre (3)



Label des points

- L'algorithme de séparation/évaluation labellise tous les points de la région réalisable, certains de manière explicite (les nœuds de l'arbre), et d'autres implicitement.
- Par exemple, le point ($x_1 = 2, x_2 = 3$) est contenu dans le sous-problème 3, dont l'optimum est en $(3, 3)$, il est donc inutile de l'évaluer.
- L'algorithme divise pour régner, et converge nécessairement du au fait qu'il élimine toujours des points à chaque étape de séparation, alors que ceux-ci sont en nombre fini.

Règles de stérilisation de sommets

1. On dit qu'un sommet de l'arbre qu'il est inutile de séparer est *stérilisé*. Il y a plusieurs possibilités pour cela :
   1.1 Le sommet est associé à une solution réalisable candidate (séparer ne l'améliorera pas)
   1.2 Le sommet est associé à un sous-problème non-réalisable.
   1.3 La valeur z optimal du sous-problème associé est inférieure (pour un problème max, et vice-versa pour un problème min) à la borne inférieure courante.
2. Un sous-problème peut-être éliminé directement si
   2.1 Le sous-problème est non-réalisable
   2.2 La borne inférieure courante est supérieure à la valeur z du sous-problème.

Retour en arrière

- On peut résoudre les problèmes de séparation-évaluation soit en largeur d'abord (on évalue tous les sous-problèmes à un niveau avant de séparer au niveau inférieur : dans l'exemple on commencerait à évaluer (3) avant (4) ou (5)).
- Ou alors comme dans l'exemple en profondeur d'abord comme dans l'exemple.
- La pratique montre que l'exploration en profondeur d'abord fonctionne mieux, en particulier avec une bonne fonction d'estimation comme la relaxation LP (voir plus loin) : on divise pour régner plutôt que d'explorer systématiquement.
- ATTENTION : séparation-évaluation n'est pas glouton : on effectue des retours en arrière (backtrack).

Résoudre les problèmes mixtes

- On ne sépare que sur les variables entières !

Généralisation de sep/eval

- La relaxation LP n'est qu'une heuristique !
- Elle peut être remplacée par une plus performante
- Exemple : algorithme du knapsack (voir en TD).
- Lien avec l'algorithme $A^{\star}$ d'exploration combinatoire.

Knapsack

- Rappel : un knapsack (problème de sac à dos) est un problème à une seule contrainte.

$$\begin{aligned} \max z &= \textstyle\sum_i c_i x_i \\ \textstyle\sum_i a_i x_i &\leq b \end{aligned}$$

avec $x_i \in \{0, 1\}$
- Séparer sur x_i donne une branche $x_i = 0$ et une branche $x_i = 1$.
- On peut résoudre la relaxation LP par inspection en remarquant qu'on peut ordonner les variables par rapport $\frac{c_i}{a_i}$ décroissant. Les variables avec le meilleur rapport sont préférables.
- Voir TD associé.
- On peut aussi résoudre un knapsack par programmation dynamique.

Planification

- Les problèmes de planification avec dates butoir sont très courants, par exemple : optimisation de ressource, gestion temps-réel, etc.
- Illustration :

	durée	date butoir
tâche 1	6	8
tâche 2	4	4
tâche 3	5	12
tâche 4	8	16

- On lit la table de la façon suivante : la tâche 1 requiert 6 jours et doit être complétée à la fin du 8ème jour.
- Chaque jour de retard résulte en des pénalités. On doit minimiser les jours de retard.

Solution par S/E

- On propose d'utiliser les variables suivantes :

$$x_{ij} = \begin{cases} 1 & \text{si la tâche } i \text{ est en position } j \\ 0 & \text{sinon} \end{cases}$$

- On peut ici proposer de séparer en assignant d'abord la *dernière* tâche, et en évaluant le délai minimal total en faisant la somme des durées pour tâches assignées et restantes moins le délai pour la dernière tâche
- Ici, si la tâche 3 est la dernière assignée ($x_{34} = 1$), on a un délai minimal estimé de $6 + 4 + 8 + 5 - 12 = 23 - 12 = 11$.

Solution par S/E, suite

- Par la suite dans l'arbre on sépare en assignant les tâches en commençant vers la fin (dernière tâche, puis avant-dernière, etc), et on évalue en faisant la somme des délais évalués. Par exemple si la tâche 3 est choisie en dernier, puis la tâche 4 en avant-dernier, le délai total ne pourra pas être inférieur à $11 + (6 + 4 + 8 - 16) = 11 + 2 = 13$.
- On pourra vérifier que la solution optimale dans l'exemple est 2-1-3-4 avec un délai de 12 jours.

Travelling Salesperson Problem

- Notez la définition politiquement correcte...
- Rappel : on a un ensemble de villes qu'on veut toutes visiter, au coût moindre.
- Prenons la matrice de coût suivante, où c_{ij} représente le coût d'aller de la ville i à la ville j :

	ville 1	ville 2	ville 3	ville 4	ville 5
ville 1	0	132	217	164	58
ville 2	132	0	290	201	79
ville 3	217	290	0	113	303
ville 4	164	201	113	0	196
ville 5	58	79	303	196	0

TSP (II)

- Le voyage de coût minimum est nécessairement un circuit, qui est un cycle Hamiltonien du graphe associé.
- On peut le résoudre par E&S, pour cela, il faut un moyen d'évaluer et un moyen de séparer. On peut par exemple transformer TSP en problème d'affectation avec la matrice de coût c : soit x_{ij} un ensemble de variables binaires, indiquant que si $x_{ij} = 1$ alors le voyage de la ville i à la ville j est effectué, et 0 sinon.
- Dans ce cas, si on les variables $x_{12} = x_{24} = x_{45} = x_{53} = x_{31} = 1$, alors on a réalisé le circuit 1-2-4-5-3-1.
- Effectivement, si le problème d'affectation génère un circuit, alors il est optimal (pourquoi ?), mais l'affectation peut ne pas générer de circuits, par exemple on pourrait obtenir la solution $x_{15} = x_{21} = x_{34} = x_{43} = x_{52} = 1$, qui est une solution avec 2 sous-circuits non connexes.

TSP (III)

- On doit aussi imposer $i \neq j$, ce qui peut se faire en imposant un coût $c_{ii} = M$ élevé.
- On peut par exemple partir de la solution d'affectation, puis séparer en éliminant des sous-circuits. Pour éliminer un sous-circuit, il suffit d'imposer un coût élevé M à une des arêtes d'un des sous-circuits.
- On peut choisir, pour une arête joignant la ville i à la ville j, d'imposer un coût élevé dans un sens ou dans l'autre : on sépare sur ce choix.
- Par exemple, avec la table donnée (modifiée pour un coût c_{ii} élevé), la première solution d'affectation est $x_{15} = x_{21} = x_{34} = x_{43} = x_{52} = 1$.
- On a deux sous-circuits : 1-5-2-1 et 3-4-3. On peut décider d'éliminer le sous-circuit 3-4-3, soit en imposant $c_{34} = M$, soit $c_{43} = M$. (Q : pourquoi est-ce légitime ?)

TSP (IV)

- Par S/E on vérifie que l'on converge vers la solution $x_{15} = x_{24} = x_{31} = x_{43} = x_{52} = 1$ soit le tour 1-5-2-4-3-1 pour un coût $z = 668$.

Résolution des problèmes binaires par énumération implicite

- Les problèmes binaires (BP) forment une très grande classe de problèmes en IP.
- Remarque : on peut exprimer tout IP en BP (par décomposition sur puissances de 2)
- Il existe des méthodes de S/E pour BP particulières, par exemple la méthode d'énumération implicite : en chaque branche les branches supérieures représentent les variables fixes. On branche sur la variable qui améliore la fonction de coût le plus (comme dans le knapsack), tant que la solution est faisable.
- Même si la solution n'est pas faisable, elle génère un coût d'évaluation.

Résolution des problèmes IP Solution par séparation-évaluation Algorithme général Formulation de problèmes intéressants Con

Sudoku

Comme promis.

Conclusion

- Optimisation par PL associée avec une PI ou PM par S/E donne une méthodologie génerale pour résoudre les PI/PM.
- Un intérêt majeur de S/E est que la convergence est garantie en temps fini, et que toute solution réalisable intermédiaire est associée avec une borne inférieure et supérieure pour le coût optimal.
- Dans certains cas on a intérêt à exhiber une meilleure fonction d'évaluation et de séparation, mais la méthode générale reste valide.
- De la qualité de ces fonctions et de la formulation dépend le résultat.
- Lien avec heuristiques, $A^\star$, algorithmes sur les graphes, etc.
- En général les problèmes solubles par S/E sont difficiles, mais parfois ce n'est pas le cas.

Problèmes de transport

formulation des problèmes d'affectation

Hugues Talbot

Laboratoire A2SI

31 mars 2009

Problèmes de Transport Solution des problèmes de transport Problèmes d'affectation Problème de transbordement Conclusion

Plan

Problèmes linéaires particuliers : problèmes de transport

- Certains problèmes en programmation linéaire ont une structure particulière que l'on peut exploiter ;
- On peut les résoudre comme d'habitude par un simplexe, mais on peut aussi les résoudre plus simplement et plus efficacement.
- Certains de ces problèmes sont formulés en entier. La solution est en entier aussi, mais la résolution n'est pas plus difficile.
- Le mieux est de donner un exemple

Distribution d'électricité

Soit un série de villes alimentées en électricité par des centrales. La situation est résumée par la table suivante :

	Cité 1	Cité 2	Cité 3	Cité 4	Puissance fournie (GWh)
Centrale 1	€8	€6	€10	€9	35
Centrale 2	€9	€12	€13	€7	50
Centrale 3	€14	€9	€16	€5	40
Demande (GWh)	45	20	30	30	

Ici, les coût au milieu de la matrice sont ceux de production pour 1GWh.
Formulez le problème pour minimiser le coût pour alimenter toutes les villes.

Formulation

- x_{ij} = nombre de GWh produits à la centrale i et envoyé à la cité j.
- Coût d'acheminement depuis les centrales = coût total =

$$\begin{array}{rrrrrrrrr} z = & & 8x_{11} & + & 6x_{12} & + & 10x_{13} & + & 9x_{14} \\ & + & 9x_{21} & + & 12x_{22} & + & 13x_{23} & + & 7x_{24} \\ & + & 14x_{31} & + & 9x_{32} & + & 16x_{33} & + & 5x_{34} \end{array}$$

Formulation : contraintes

- Contraintes de production

$$\begin{array}{ccccccccc} x_{11} & + & x_{12} & + & x_{13} & + & x_{14} & \leq & 35 \\ x_{21} & + & x_{22} & + & x_{23} & + & x_{24} & \leq & 50 \\ x_{31} & + & x_{32} & + & x_{33} & + & x_{34} & \leq & 40 \end{array}$$

- contraintes de consommation

$$\begin{array}{ccccccc} x_{11} & + & x_{21} & + & x_{31} & \geq & 45 \\ x_{12} & + & x_{22} & + & x_{32} & \geq & 40 \\ x_{13} & + & x_{23} & + & x_{33} & \geq & 30 \\ x_{14} & + & x_{24} & + & x_{34} & \geq & 30 \end{array}$$

- Plus les contraintes habituelles ($x_{ij} \geq 0$)

Résolution

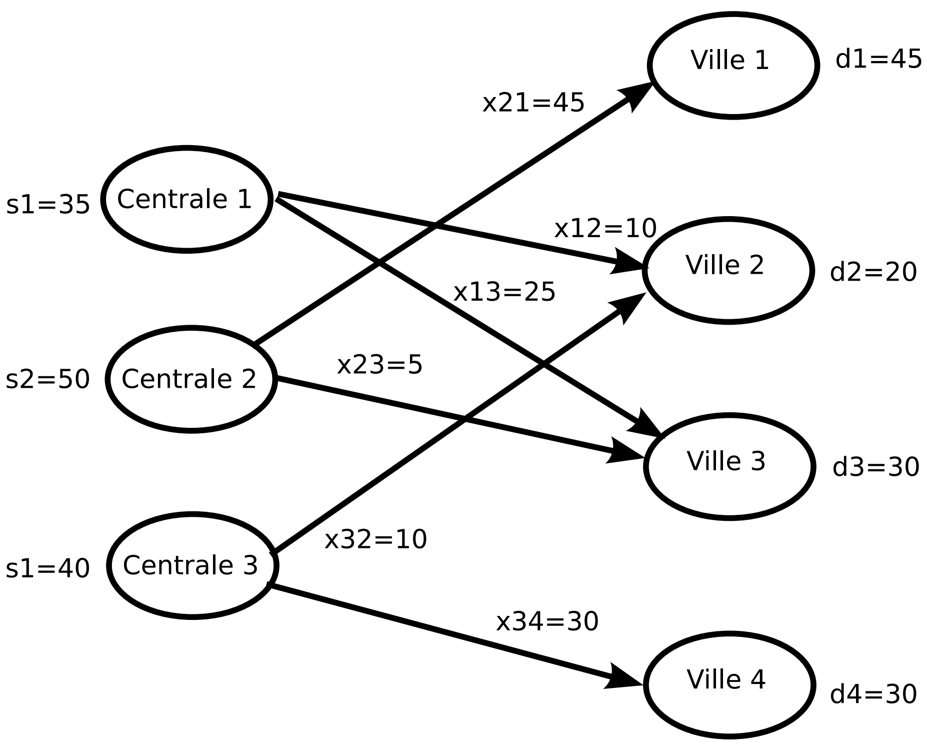
- C'est un problème de PL standard
- On peut résoudre par un simplexe (pas à la main...).
- On trouve le résultat suivant :

x_{12}	x_{13}	x_{21}	x_{23}	x_{32}	x_{34}
10	25	45	5	10	30

Pour un coût total de 1020.

Problèmes de Transport Solution des problèmes de transport Problèmes d'affectation Problème de transbordement Conclusion

Solution sous forme graphique



Forme générale

La forme générale d'un problème de transport est la suivante :

$$\min \sum_{i=1}^{i=m} \sum_{j=1}^{j=n} c_{ij} x_{ij}$$

$$s.t \sum_{j=1}^{j=n} x_{ij} \leq s_i, i \in \{1, \ldots, m\} \qquad \text{(contraintes d'offre)}$$

$$\sum_{i=1}^{i=m} x_{ij} \geq d_j, j \in \{1, \ldots, n\} \qquad \text{(contraintes de demande)}$$

Terminologie

- Si le problème est une maximisation, c'est toujours un problème de transport.
- Si on a

$$\sum_{i=1}^{i=m} s_i = \sum_{j=1}^{j=n} d_j,$$

  le problème est dit *équilibré*.
  L'exemple donné est bien équilibré.
- Dans un problème équilibré, toutes les contraintes doivent être des égalités (pourquoi ?).

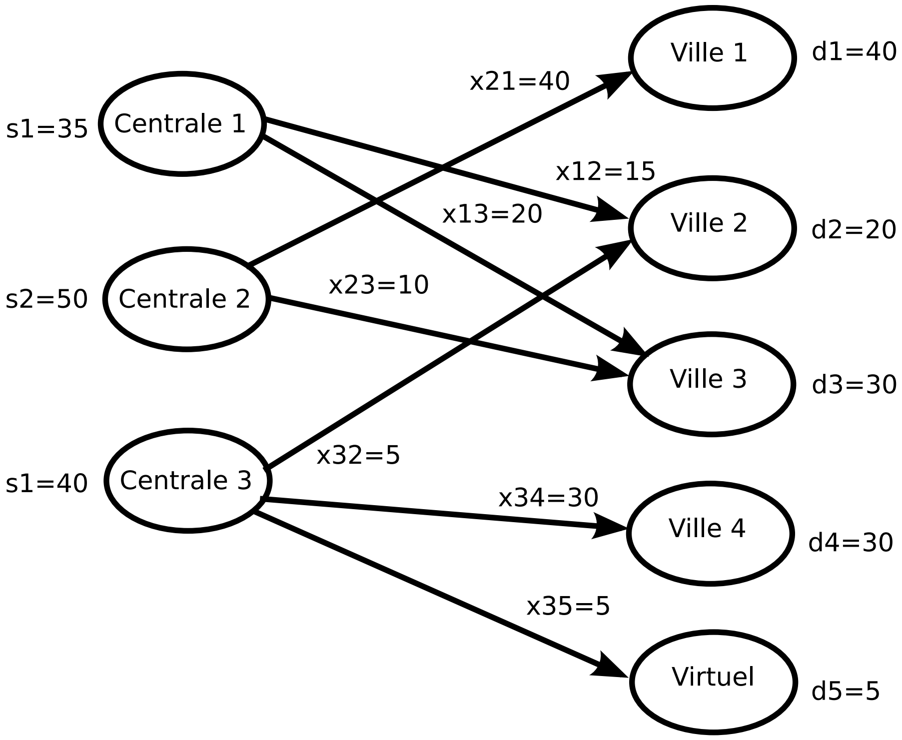
Problèmes équilibrés

- Il est préférable de considérer les problèmes équilibrés. En effet, on montrera qu'il est *relativement* facile de trouver une solution de base réalisable pour ces problèmes.
- De même, les opérations du simplexe dans le cas de problèmes de transport équilibrés se réduisent à des additions et soustractions.

Rendre un problème équilibré

- Pour équilibrer un problème de transport pour lequel il y a trop d'offre, il suffit de créer un *point de demande virtuel* dont la demande correspond à l'offre excédentaire, et pour lequel les coûts de transport sont nuls.
- La demande transportée vers le point virtuel correspond à la capacité non utilisée. De manière naturelle, c'est le point d'offre pour lequel les coûts de transport sont les (question : plus ? moins ?) élevés qui enverra sa capacité vers le lien virtuel.
- Exemple, dans le cas précédent de livraison d'électricité, en supposant que la demande pour la cité 1 soit réduite à 40 GWh. On trouve un excès de 5 GWh, qu'on peut allouer à un point de demande virtuel.
- Notons que la solution optimale est assez différente dans ce cas.

Solution sous forme graphique du cas non équilibré



Représentation sous forme de tableau

- On peut facilement représenter un problème de transport sous forme de tableau :

	Ville 1		Ville 2		Ville 3		Ville 4		Offre
centrale 1	0	8	10	6	25	10	0	9	35
centrale 2	45	9	0	12	5	13	0	7	50
centrale 3	0	14	10	9	0	16	30	5	40
Demande	45		20		30		30		

- On note que les valeurs s'additionne en lignes et en colonnes.

Equilibrage lorsque la demande excède l'offre

- Lorsque la demande excède l'offre, il n'y a pas de solution réalisable (exemple : réduisons la capacité de la centrale 1 à 30 GWh).
- Parfois, la modélisation permet d'avoir de la demande non satisfaite, souvent en ajoutant une pénalité.
- Exemple : production d'eau

Problème de production d'eau

- Deux réservoirs sont prévus pour alimenter 3 villes en eau potable. Chacun des réservoirs peut produire 50 000 m^3 d'eau par jour.
- La demande de chacune des villes est de 40 000 m^3/j
- Si les réservoirs n'arrivent pas à fournir suffisamment d'eau, il y a une pénalité par 1000 m^3 : €20 pour la ville 1, €22 pour la ville 2 et €23 pour la ville 3.
- Les coûts de transport par 1000 m^3 sont résumés ici :

De à	Ville 1	Ville 2	Ville 3
Réservoir 1	€7	€8	€10
Réservoir 2	€9	€7	€8

Solution

	Ville 1		Ville 2		Ville 3		Offre
Réservoir 1	20	$\boxed{7}$	30	$\boxed{8}$	0	$\boxed{10}$	50
Réservoir 2	0	$\boxed{9}$	10	$\boxed{7}$	40	$\boxed{8}$	50
Virtuel	20	$\boxed{20}$	0	$\boxed{22}$	0	$\boxed{23}$	20
Demande	40		40		40		

Modélisation des problèmes d'inventaire

Sur un exemple

- L'entreprise BellesVoiles fabrique des voiles pour bateaux. Elle a son carnet de commande pour les 4 prochains trimestres :

	1er	2eme	3eme	4eme
commandes	40	60	75	25

- BV doit fournir à temps. Elle possède un inventaire de 10 voiles et doit décider de combien de voiles produire par trimestre au début de chacun d'entre eux. On suppose que seules les voiles produites durant un trimestre peuvent être vendues.
- Chaque trimestre, BV peut produire jusqu'à 40 voiles à un coût de €400 par voile, ou bien, en payant ses employés des heures supplémentaires, jusqu'à 40 voiles à un coût de 450 chacune.
- A la fin de chaque trimestre, un coût d'inventaire de €20 doît être appliqué à chaque invendu.
- On veut minimiser les coûts et produire à temps.

Production pour les voiles

- Points d'offre :
  1. Inventaire initial ($s_1 = 10$)
  2. Production du premier trimestre : Normale ($s_2 = 40$), heures sup ($s_3 = 40$).
  3. Production du second trimestre : Normale ($s_4 = 40$), heures sup ($s_5 = 40$).
  4. Production du troisième trimestre : Normale ($s_6 = 40$), heures sup ($s_7 = 40$).
  5. Production du quatrième trimestre : Normale ($s_8 = 40$), heures sup ($s_9 = 40$).

  Total de l'offre : 330

Consommation pour les voiles

- Points de demande :
  1. Demande du premier trimestre ($d_1 = 40$)
  2. Demande du second trimestre ($d_2 = 60$)
  3. Demande du troisième trimestre ($d_3 = 75$)
  4. Demande du quatrième trimestre ($d_4 = 25$)
  5. Demande virtuelle pour équilibrer ($d_5 = 330 - 200 = 130$).
- Remarque : il faut empêcher de produire une voile au 2e trimestre pour remplir la production du 1er ! ce type de transport doit être impossible.

Tableau pour les voiles

	Trimestre 1	Trimestre 2	Trimestre 3	Trimestre 4	Virtuel	Offre
Initial	$\boxed{0}$	$\boxed{20}$	$\boxed{40}$	$\boxed{60}$	$\boxed{0}$	
T1 TN	$\boxed{400}$	$\boxed{420}$	$\boxed{440}$	$\boxed{460}$	$\boxed{0}$	
T1 HS	$\boxed{450}$	$\boxed{470}$	$\boxed{490}$	$\boxed{510}$	$\boxed{0}$	
T2 TN	$\boxed{M}$	$\boxed{400}$	$\boxed{420}$	$\boxed{440}$	$\boxed{0}$	
T2 HS	$\boxed{M}$	$\boxed{450}$	$\boxed{470}$	$\boxed{490}$	$\boxed{0}$	
T3 TN	$\boxed{M}$	$\boxed{M}$	$\boxed{400}$	$\boxed{420}$	$\boxed{0}$	
T3 HS	$\boxed{M}$	$\boxed{M}$	$\boxed{450}$	$\boxed{470}$	$\boxed{0}$	
T4 TN	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{400}$	$\boxed{0}$	
T4 HS	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{450}$	$\boxed{0}$	
Demande	40	60	75	25	130	

Solution pour les voiles

	Trimestre 1	Trimestre 2	Trimestre 3	Trimestre 4	Virtuel	Offre
Initial	10 $\boxed{0}$	$\boxed{20}$	$\boxed{40}$	$\boxed{60}$	$\boxed{0}$	10
T1 TN	30 $\boxed{400}$	10 $\boxed{420}$	$\boxed{440}$	$\boxed{460}$	$\boxed{0}$	40
T1 HS	$\boxed{450}$	$\boxed{470}$	$\boxed{490}$	$\boxed{510}$	40 $\boxed{0}$	40
T2 TN	$\boxed{M}$	40 $\boxed{400}$	$\boxed{420}$	$\boxed{440}$	$\boxed{0}$	40
T2 HS	$\boxed{M}$	10 $\boxed{450}$	$\boxed{470}$	$\boxed{490}$	30 $\boxed{0}$	40
T3 TN	$\boxed{M}$	$\boxed{M}$	40 $\boxed{400}$	$\boxed{420}$	$\boxed{0}$	40
T3 HS	$\boxed{M}$	$\boxed{M}$	35 $\boxed{450}$	$\boxed{470}$	5 $\boxed{0}$	40
T4 TN	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	25 $\boxed{400}$	15 $\boxed{0}$	40
T4 HS	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{450}$	40 $\boxed{0}$	40
Demande	40	60	75	25	130	

Trouver une base de départ

- Soit un problème de transport avec m points d'offre et n points de demande. C'est un problème avec $m + n$ contraintes d'égalité.
- Il est difficile de trouver une SBR initiale dans le cas des égalités strictes (pourquoi ?).
- Une remarque est très importante : dans les problèmes de transport à $m + n$ égalités, une de ces égalités est redondante. autrement dit, si on trouve un ensemble de x_{ij} qui satisfait toutes les contraintes sauf une, alors la dernière est également satisfaite.

Variables indépendantes

- par exemple, dans le cas de la distribution d'électricité, si on ignore la première égalité, on voit qu'elle est tout de même satisfaite par la solution.
- Dans les $m + n - 1$ contraintes restantes, on ne peut pas se contenter de prendre n'importe quelle collection de $n + m - 1$ variables comme base de départ. On peut tomber sur une matrice de rang trop faible.

Boucles et bases

- Une séquence de au moins 4 cellules d'un tableau est une boucle si et seulement si :
  - toute paire consécutive de cellules sont soit sur la même ligne, soit sur la même colonne
  - aucun triplet de cellules sont sur la même ligne ou colonne
  - la première et la dernière cellule sont consécutive (soit sur la même ligne, soit sur la même colonne)
- On a le théorème suivant :
  Dans un problème de transport équilibré avec m producteurs et n consommateurs, les cellules correspondant à un ensemble de $m + n - 1$ variables forment une solution de base si et seulement si l'ensemble des cellules correspondant ne contient pas de boucles.

Problèmes de Transport Solution des problèmes de transport Problèmes d'affectation Problème de transbordement Conclusion

Méthodes pour trouver une SBR initiale

Il y a trois méthodes classiques

1. La méthode du coin supérieur gauche ;
2. La méthode du coût minimum ;
3. La méthode de VOGEL.

Exemple de problème d'affectation

- Une fabrique M a 4 machines et 4 tâches à compléter
- Chaque machine doit lui voir assigner une tâche. Le temps de mise en œuvre est donné par la table suivante :

	T1	T2	T3	T4
Machine 1	14	5	8	7
Machine 2	2	12	6	5
Machine 3	7	8	3	9
Machine 4	2	4	6	10

- La fabrique veut minimiser le temps total de mise en œuvre.
- Formuler et résoudre

Formulation des problèmes d'affectation

- Les problèmes d'affectation sont des problèmes de transport équilibrés pour lesquels chaque producteur et consommateur ont une valeur de 1.
- Si toutes les valeurs dans le tableau de transport sont entières, la solution le sera aussi. On peut donc ignorer cette contrainte si elle surgit.
- En passant : peut on avoir un problème d'affectation $m \times n$ avec $n \neq m$? si oui, à quelle situation avons nous affaire, et que faire ?

Trouver une solution

- Dans n'importe quel ensemble de variables de bases pour un problème d'affectation de taille $m \times m$, on aura toujours m variable qui valent 1 et $m - 1$ variables qui valent 0 (pourquoi ?).
- On peut trouver une SBR initiale et résoudre par le simplexe des transports, mais les variables de base des problèmes d'affectation sont très dégénérées et le simplexe n'est pas bien adapté.

La méthode "Hongroise"

1. Trouver l'élément minimum dans chaque ligne de la matrice $m \times m$. Construire une nouvelle matrice en soustrayant de chaque coût le minimum dans sa ligne ; Pour cette nouvelle matrice, trouver le coût minimum dans chaque colonne. Construire une nouvelle matrice en soustrayant dans chaque colonne son minimum.
2. Tracer le nombre minimum de lignes (horizontales ou verticales) pour couvrir tous les zéros dans cette nouvelle matrice (appellée la matrice des coûts réduits). Si moins m lignes sont nécessaires, passer à l'étape 3.
3. Trouver le plus petit élément non-nul k dans la matrice des coûts réduits, qui ne soit pas couvert par une ligne à l'étape 2. Soustraire k de chaque élément non recouvert, puis ajouter k à tous les éléments recouverts par 2 lignes, et retourner à l'étape 2.

Résolution par la méthode Hongroise

1- Minimum par lignes

	Tâche 1	Tâche 2	Tâche 3	Tâche 4	Min
Machine 1	14	5	8	7	5
Machine 2	2	12	6	5	2
Machine 3	7	8	3	9	3
Machine 4	2	4	6	10	2

Résolution par la méthode Hongroise

2- Minimum par colonnes

	Tâche 1	Tâche 2	Tâche 3	Tâche 4
Machine 1	9	0	3	2
Machine 2	0	10	4	3
Machine 3	4	5	0	6
Machine 4	0	2	4	8
Minimum	0	0	0	2

Résolution par la méthode Hongroise

3- lignes

	Tâche 1	Tâche 2	Tâche 3	Tâche 4
Machine 1	+ 9	– 0	– 3	– 0
Machine 2	| 0	10	4	1
Machine 3	+ 4	– 5	– 0	– 4
Machine 4	| 0	2	4	6

Résolution par la méthode Hongroise

3- Minimum par cellules non couvertes : 1

	Tâche 1	Tâche 2	Tâche 3	Tâche 4
Machine 1	+ 10	– 0	– 3	+ 0
Machine 2	| 0	9	3	| 0
Machine 3	+ 5	– 5	– 0	+4
Machine 4	| 0	1	3	| 5

Choix de la base optimale

- A la fin de l'algorithme Hongrois, on a au moins m zéros couverts dans la matrice
- Il existe un et un seul ensemble de variables constituées de zéros couverts qui forme un SBR
- Ce SBR est la base optimale pour le problème d'affectation
- Ici : $x_{41} = x_{12} = x_{33} = x_{24} = 1$.

Justification intuitive

- Si une constante k est ajoutée à chaque ligne ou colonne dans un problème de transport équilibré, la solution optimale n'est pas changée.Cela revient à ajouter la constante k au coût, puisque par exemple $\sum_{j=1}^{m} x_{1j} = 1$
- De même, l'étape 3 de la méthode Hongroise ne change pas l'optimum, car elle revient à faire simultanément :
  - ajouter k à chaque coût couvert par une ligne horizontale ;
  - soustraire k à chaque coût non-couvert par une ligne verticale.
- Étape 1 crée au moins un zéro par ligne ou colonne. Étape 3 crée au moins un zéro supplémentaire à chaque fois.
- Ces deux étapes opèrent tout en gardant les coût non-négatifs.
- Au bout du compte, l'optimum est le même que pour le problème initial, et il est forcément constitué de coûts nuls.

Exemple de problème de transbordement

- Soit l'entreprise W, qui fabrique des jouets, l'une à Montpellier, l'autre à Douais. Celle de Montpellier peut en fabriquer 150 par jour, celle de Douais 200.
- Les jouets sont envoyé par la route aux magasins à Lyon et Brest. Les clients dans ces villes achètent 130 jouets.
- Du fait des coûts de transports moins élevés par rail, il peut être moins cher de faire passer les jouets par Nevers et/ou Castres. Les coûts d'acheminement sont les suivants :

	M	D	N	C	L	B
M	0	-	8	13	25	28
D	-	0	15	12	26	25
N	-	-	0	6	16	17
C	-	-	6	0	14	16
L	-	-	-	-	0	-
B	-	-	-	-	-	0

Transformation en problème de transport

Conclusion

- Les problèmes de transport, affectaction et transbordement sont des cas particuliers de LP, qu'on ne résout pas par le simplexe habituel.
- Il existe une méthode de résolution plus simple, non matricielle.
- Si les coût sont entiers, la solution est également entière, donc si on peut formuler un problème sous forme de transport, la solution en entier est également facilement calculable.

Problèmes de transport et transbordement

Résolution

Hugues Talbot

Laboratoire A2SI

9 avril 2009

Plan

Introduction

- Rappel : les problèmes de transport sont des problèmes de programmation linéaires associant des producteurs et des consommateurs ;
- On peut toujours équilibrer un problème de transport de telle manière que toute la production soit consommée, au prix de nœuds supplémentaires ;
- Les problèmes de transport se résolvent plus facilement que les PL standards. Il n'y a pas d'inversion de matrice, les seules opérations sont des additions et soustractions
- Les problèmes de transports entiers ne sont pas plus difficiles que les autres.

Rappels

- On peut représenter un problème de transport dans un tableau ;
- Un problème a m producteurs et n consommateurs est au plus de rang $m + n - 1$ (Q : pourquoi ?) ;
- Un problème de transport équilibré n'a que des égalités (Q : pourquoi ?)
- Un problème avec uniquement des égalités est souvent plus difficile à démarrer (c-à-d trouver une base réalisable initiale) que les problèmes à égalités (Q : pourquoi ?)

Exemple

Problème de transport

			4
			5
3	2	4	

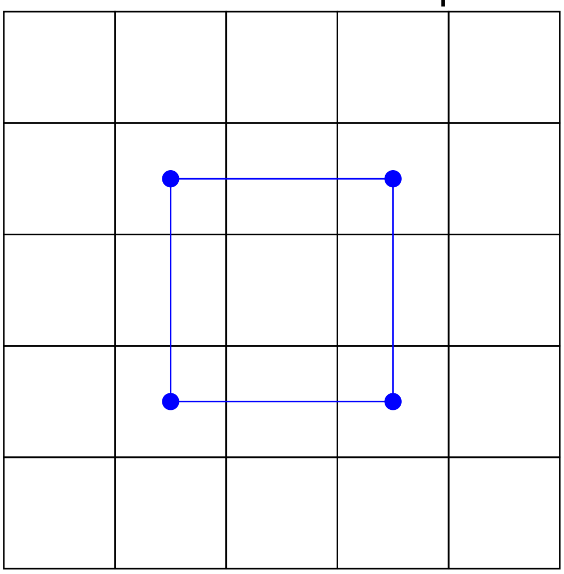
Problème PL équivalent

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 3 \\ 2 \\ 4 \end{bmatrix}$$

- On doit éliminer une contrainte (p.ex. la première ligne) pour en faire un pb de rang $m + n - 1 = 4$
- Trouver une base de départ n'est pas simple. Par exemple $\{x_{11}, x_{12}, x_{21}, x_{22}\}$ ne marche pas.

Notion de boucle

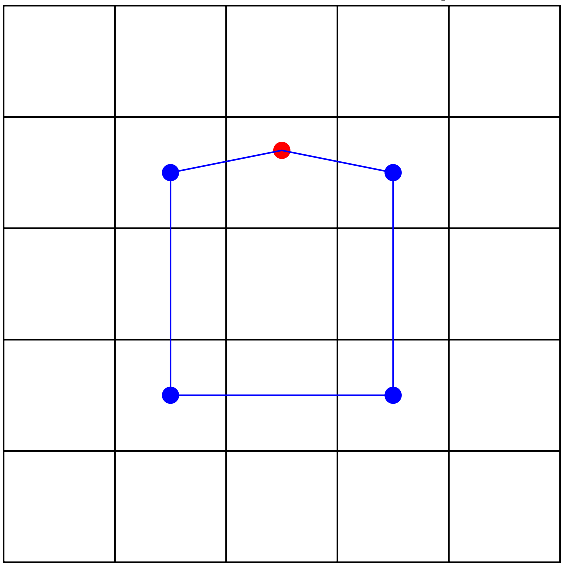
Une boucle est une séquence de 4 cellules au moins, telle que :



- Deux cellules consécutives sont dans la même ligne ou même colonne ;

Notion de boucle

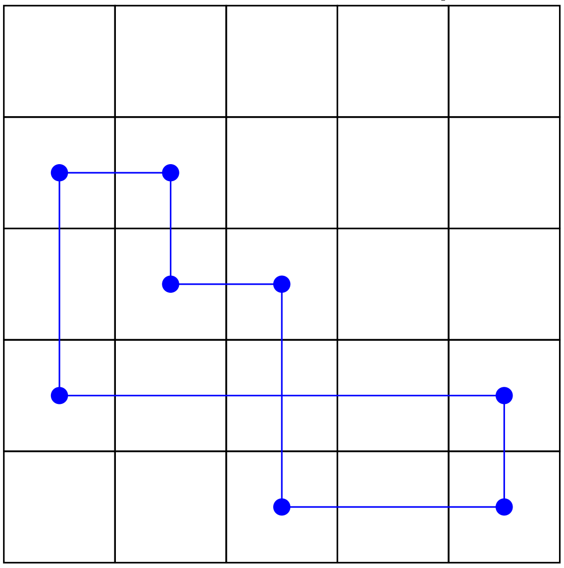
Une boucle est une séquence de 4 cellules au moins, telle que :



- Deux cellules consécutives sont dans la même ligne ou même colonne ;
- Toute suite de trois cellules consécutives ne sont *jamais* dans la même ligne ou colonne ;

Notion de boucle

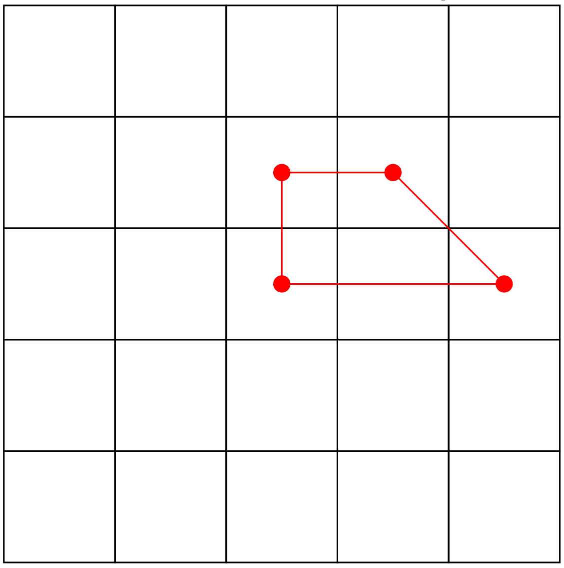
Une boucle est une séquence de 4 cellules au moins, telle que :



- Deux cellules consécutives sont dans la même ligne ou même colonne ;
- Toute suite de trois cellules consécutives ne sont *jamais* dans la même ligne ou colonne ;
- La dernière cellule dans la séquence a une ligne ou une colonne en commun avec la première

Notion de boucle

Une boucle est une séquence de 4 cellules au moins, telle que :



- Deux cellules consécutives sont dans la même ligne ou même colonne ;
- Toute suite de trois cellules consécutives ne sont *jamais* dans la même ligne ou colonne ;
- La dernière cellule dans la séquence a une ligne ou une colonne en commun avec la première

Théorème des boucles

Théorème

Soit un problème de transport avec m producteurs et n consommateurs. Les cellules qui correspondent à un ensemble de $m + n - 1$ variables ne contiennent aucune boucle si et seulement si les $m + n - 1$ variables forment une solution de base.

Démonstration.

Ce théorème découle du fait qu'un ensemble de $m + n - 1$ cellules ne contiennent aucune boucle si et seulement si les $m + n - 1$ colonnes de A qui correspondent à ces cellules sont linéairement indépendantes. $\square$

Méthodes pour trouver une SBR initiale

Il y a trois méthodes classiques

1. La méthode du coin supérieur gauche ;
2. La méthode du coût minimum ;
3. La méthode de VOGEL.

Introduction | Solution des problèmes de transport | Problème de transbordement | Conclusion

La méthode du coin supérieur gauche (MCSG)

- On commence en haut à gauche par x_{11}, et on augmente x_{11} autant que possible ;

				5
				1
				3
2	4	2	1	

La méthode du coin supérieur gauche (MCSG)

2				3
				1
				3
X	4	2	1	

- On commence en haut à gauche par x_{11}, et on augmente x_{11} autant que possible ;
- On élimine du tableau la ligne ou la colonne saturée, on diminue de x_{11} la ligne ou colonne non saturée ;

La méthode du coin supérieur gauche (MCSG)

2	3			X
				1
				3
X	1	2	1	

- On commence en haut à gauche par x_{11}, et on augmente x_{11} autant que possible ;
- On élimine du tableau la ligne ou la colonne saturée, on diminue de x_{11} la ligne ou colonne non saturée ;
- On continue cette procédure récursivement sur le reste du tableau ;

La méthode du coin supérieur gauche (MCSG)

2	3			X
	1			X
				3
X	0	2	1	

- On commence en haut à gauche par x_{11}, et on augmente x_{11} autant que possible ;
- On élimine du tableau la ligne ou la colonne saturée, on diminue de x_{11} la ligne ou colonne non saturée ;
- On continue cette procédure récursivement sur le reste du tableau ;
- Dans le cas où une augmentation sature la ligne et la colonne en même temps, on choisit d'éliminer seulement soit la ligne, soit la colonne ;

La méthode du coin supérieur gauche (MCSG)

2	3			X
	1			X
	0			3
X	X	2	1	

- On commence en haut à gauche par x_{11}, et on augmente x_{11} autant que possible ;
- On élimine du tableau la ligne ou la colonne saturée, on diminue de x_{11} la ligne ou colonne non saturée ;
- On continue cette procédure récursivement sur le reste du tableau ;
- Dans le cas où une augmentation sature la ligne et la colonne en même temps, on choisit d'éliminer seulement soit la ligne, soit la colonne ;
- Une saturation à zéro donne une base initiale dégénérée, comme ici ;

La méthode du coin supérieur gauche (MCSG)

2	3			X
	1			X
	0	2		1
X	X	X	1	

- On commence en haut à gauche par x_{11}, et on augmente x_{11} autant que possible ;
- On élimine du tableau la ligne ou la colonne saturée, on diminue de x_{11} la ligne ou colonne non saturée ;
- On continue cette procédure récursivement sur le reste du tableau ;
- Dans le cas où une augmentation sature la ligne et la colonne en même temps, on choisit d'éliminer seulement soit la ligne, soit la colonne ;
- Une saturation à zéro donne une base initiale dégénérée, comme ici ;

La méthode du coin supérieur gauche (MCSG)

2	3			X
	1			X
	0	2	1	X
X	X	X	X	

- On commence en haut à gauche par x_{11}, et on augmente x_{11} autant que possible ;
- On élimine du tableau la ligne ou la colonne saturée, on diminue de x_{11} la ligne ou colonne non saturée ;
- On continue cette procédure récursivement sur le reste du tableau ;
- Dans le cas où une augmentation sature la ligne et la colonne en même temps, on choisit d'éliminer seulement soit la ligne, soit la colonne ;
- Une saturation à zéro donne une base initiale dégénérée, comme ici ;
- La dernière case sature à la fois sa ligne et sa colonne.

Solution de base initiale $\{x_{11} = 2, x_{12} = 3, x_{22} = 1, x_{32} = 0, x_{33} = 2, x_{34} = 1\}$

Éléments de justification pour la MCSG

- Toutes les variables sont positives ou nulles ;
- La méthode du CSG assure que $m + n - 1$ variables sont assignées ;
- La dernière affectation sature deux contraintes, donc $m + n$ contraintes sont satisfaites. Autrement dit toutes les contraintes sont satisfaites (puisque toutes les lignes et colonnes sont saturées) ;
- La méthode du CSG assure que les variables assignées ne peuvent pas former de boucle ;
- Les variables assignées forment donc une solution de base réalisable par le théorème des boucles.

Faiblesses de la MCSG

- La méthode du CSG donne bien un SBR, mais elle peut-être très loin de l'optimal ;
- La méthode du CSG a tendance à donner des SBR dégénérées (avec des variables de base à zéro) ;
- Elle ne tient pas compte du tout du coût.
- Pour tenter de pallier ces problèmes, nous allons explorer deux autres méthodes. La première est celle du coût minimum.

La méthode du coût minimum

2	3	5	6	5
2	1	3	5	10
3	8	4	6	15
12	8	4	6	

- On commence par chercher la variable x_{ij} avec le coût de transport minimum ;

La méthode du coût minimum

2	3	5	6	5
2	1 8	3	5	2
3	8	4	6	15
12	X	4	6	

- On commence par chercher la variable x_{ij} avec le coût de transport minimum ;
- On sature sa valeur, et on ferme la ligne ou colonne correspondante ;

La méthode du coût minimum

2	3	5	6	5
2 / 2	1 / 8	3	5	X
3	8	4	6	15
10	X	4	6	

- On commence par chercher la variable x_{ij} avec le coût de transport minimum ;
- On sature sa valeur, et on ferme la ligne ou colonne correspondante ;
- Répéter la procédure avec les cases non fermées ;

La méthode du coût minimum

2 / 5	3	5	6	X
2 / 2	1 / 8	3	5	X
3	8	4	6	15
5	X	4	6	

- On commence par chercher la variable x_{ij} avec le coût de transport minimum ;
- On sature sa valeur, et on ferme la ligne ou colonne correspondante ;
- Répéter la procédure avec les cases non fermées ;
- Si une variable satisfait à la fois la contrainte de ligne et de colonne, ne fermer qu'une d'entre elles ;

La méthode du coût minimum

5 (2)	(3)	(5)	(6)	X
2 (2)	8 (1)	(3)	(5)	X
5 (3)	(8)	(4)	(6)	10
X	X	4	6	

- On commence par chercher la variable x_{ij} avec le coût de transport minimum ;
- On sature sa valeur, et on ferme la ligne ou colonne correspondante ;
- Répéter la procédure avec les cases non fermées ;
- Si une variable satisfait à la fois la contrainte de ligne et de colonne, ne fermer qu'une d'entre elles ;

La méthode du coût minimum

5 (2)	(3)	(5)	(6)	X
2 (2)	8 (1)	(3)	(5)	X
5 (3)	(8)	4 (4)	(6)	6
X	X	X	6	

- On commence par chercher la variable x_{ij} avec le coût de transport minimum ;
- On sature sa valeur, et on ferme la ligne ou colonne correspondante ;
- Répéter la procédure avec les cases non fermées ;
- Si une variable satisfait à la fois la contrainte de ligne et de colonne, ne fermer qu'une d'entre elles ;

La méthode du coût minimum

5 (2)	(3)	(5)	(6)	X
2 (2)	8 (1)	(3)	(5)	X
5 (3)	(8)	4 (4)	6 (6)	X
X	X	X	X	

- On commence par chercher la variable x_{ij} avec le coût de transport minimum ;
- On sature sa valeur, et on ferme la ligne ou colonne correspondante ;
- Répéter la procédure avec les cases non fermées ;
- Si une variable satisfait à la fois la contrainte de ligne et de colonne, ne fermer qu'une d'entre elles ;
- Lorsqu'il ne reste plus qu'une case, fermer sa ligne et sa colonne.

Justification de la MCM

- La solution trouvée est une SBR initiale par les mêmes arguments que pour la MCSG ;
- On peut espérer un moindre coût total de part la méthodologie.
- Ceci dit, comme l'algorithme de sélection de variables est glouton, on trouve des contre-exemples défavorables pour cette méthode :

(6)	(7)	(8)	10
(15)	(80)	(78)	15
15	5	5	

- La méthode de VOGEL est plus favorable, mais on ne la verra pas dans le cadre de ce cours.

Le simplexe des problèmes de transport

Étapes de l'algorithme

1. Si on n'est pas à l'optimum (voir plus loin), alors :
   1.1 Déterminer quelle variable doit entrer dans le système de base (voir plus loin) ;
   1.2 Trouver la boucle impliquant la nouvelle variable et un sous-ensemble des variables existantes ;
   1.3 Énumérez les variables dans la boucle à partir de la nouvelle variable prenant l'index 0 ;
   1.4 Trouver la cellule impaire dans la boucle contenant la variable avec la plus petite valeur θ ;
   1.5 Augmenter de θ toutes les variables d'indice pair dans la boucle, et réduire de θ toutes les variables d'indice impair ;
   1.6 Les valeurs des variables hors-boucle ne changent pas.
2. Retourner en 1.

Illustration sur le pb. de distribution d'électricité

On se rappelle le problème de distribution d'électricité du cours précédent :

	Ville 1	Ville 2	Ville 3	Ville 4	Offre
centrale 1	0 $\boxed{8}$	0 $\boxed{6}$	0 $\boxed{10}$	0 $\boxed{9}$	35
centrale 2	0 $\boxed{9}$	0 $\boxed{12}$	0 $\boxed{13}$	0 $\boxed{7}$	50
centrale 3	0 $\boxed{14}$	0 $\boxed{9}$	0 $\boxed{16}$	30 $\boxed{5}$	40
Demande	45	20	30	30	

Résolution du problème d'électricité

(8)	(6)	(10)	(9)	35
(9)	(12)	(13)	(7)	50
(14)	(9)	(16)	(5)	40
45	20	30	30	

- Avant initialisation par la MCSG

Résolution du problème d'électricité

35 (8)	(6)	(10)	(9)	35
10 (9)	20 (12)	20 (13)	(7)	50
(14)	(9)	10 (16)	30 (5)	40
45	20	30	30	

- Avant initialisation par la MCSG
- Après initialisation par la MCSG

Calcul des coûts réduits

- On se rappelle de la formule $\bar{\mathbf{c}}_e^T = \mathbf{c}_e^T - \mathbf{c}_b^T \mathbf{B}^{-1}\mathbf{E}$ du simplexe « normal ».
- Ici il nous faut calculer $\mathbf{c}_b^T \mathbf{B}^{-1}$, qui est un vecteur de même longueur que $\mathbf{c}_b$, c'est à dire un vecteur de $m+n-1$ éléments.
- On pose $\mathbf{c}_b^T \mathbf{B}^{-1} = [u_2 u_3 \dots u_m v_1 v_2 \dots v_n]$, où les u_i sont les contraintes de l'offre et les v_i les contraintes de la demande. Notez qu'on a abandonné une contrainte pour en garder $m+n-1$, qui est le rang du problème (voir début de ce cours).
- Le coût réduit d'une variable de base est nul, donc, pour toute variable de base x_{ij}, nous avons

$$c_{ij} = \mathbf{c}_b \mathbf{B} \mathbf{a}_{ij}$$

où c_{ij} est le coût associé à la variable x_{ij} et $\mathbf{a}_{ij}$ la colonne de $\mathbf{A}$ associée à la même variable.

Problème de PL associé au problème d'électricité

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{31} \\ x_{32} \\ x_{33} \\ x_{34} \end{bmatrix} = \begin{bmatrix} 35 \\ 50 \\ 40 \\ 45 \\ 20 \\ 30 \\ 30 \end{bmatrix}$$

NOTE : on doit éliminer la première ligne !

Illustration sur le problème d'électricité

35 (8)	(6)	(10)	(9)	35
10 (9)	20 (12)	20 (13)	(7)	50
(14)	(9)	10 (16)	30 (5)	40
45	20	30	30	

- $\bar{c}_{11} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - 8 =$

  $v_1 - 8 = 0$

Illustration sur le problème d'électricité

35 (8)	(6)	(10)	(9)	35
10 (9)	20 (12)	20 (13)	(7)	50
(14)	(9)	10 (16)	30 (5)	40
45	20	30	30	

- $v_1 - 8 = 0$
- $\bar{c}_{21} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - 9 =$

  $u_2 + v_1 - 9 = 0$

Illustration sur le problème d'électricité

35 [8]	[6]	[10]	[9]	35
10 [9]	20 [12]	20 [13]	[7]	50
[14]	[9]	10 [16]	30 [5]	40
45	20	30	30	

- $v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $\bar{c}_{22} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} - 12 =$

  $u_2 + v_2 - 12 = 0$

Introduction | Solution des problèmes de transport | Problème de transbordement | Conclusion

Illustration sur le problème d'électricité

35 [8]	[6]	[10]	[9]	35
10 [9]	20 [12]	20 [13]	[7]	50
[14]	[9]	10 [16]	30 [5]	40
45	20	30	30	

- $v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $u_2 + v_2 - 12 = 0$
- $\bar{c}_{23} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} - 13 =$

  $u_2 + v_3 - 13 = 0$

◂ □ ▸ ◂ ▸ ◂ ≡ ▸ ◂ ≡ ▸ ≡ ↶ 🔍 ↷

Illustration sur le problème d'électricité

35 (8)	(6)	(10)	(9)	35
10 (9)	20 (12)	20 (13)	(7)	50
(14)	(9)	10 (16)	30 (5)	40
45	20	30	30	

- $v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $u_2 + v_2 - 12 = 0$
- $u_2 + v_3 - 13 = 0$
- $\bar{c}_{33} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} - 16 =$

$u_3 + v_3 - 16 = 0$

Illustration sur le problème d'électricité

35 (8)	(6)	(10)	(9)	35
10 (9)	20 (12)	20 (13)	(7)	50
(14)	(9)	10 (16)	30 (5)	40
45	20	30	30	

- $v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $u_2 + v_2 - 12 = 0$
- $u_2 + v_3 - 13 = 0$
- $u_3 + v_3 - 16 = 0$

$\bar{c}_{34} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - 5 =$

$u_3 + v_4 - 5 = 0$

Illustration sur le problème d'électricité

35 (8)	(6)	(10)	(9)	35
10 (9)	20 (12)	20 (13)	(7)	50
(14)	(9)	10 (16)	30 (5)	40
45	20	30	30	

- $v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $u_2 + v_2 - 12 = 0$
- $u_2 + v_3 - 13 = 0$
- $u_3 + v_3 - 16 = 0$
- $u_3 + v_4 - 5 = 0$

On voit que si on pose $u_1 = 0$, toutes ces équations se réduisent à $u_i + v_j = c_{ij}$ pour les variables de base x_{ij}.

Illustration sur le problème d'électricité

35 (8)	(6)	(10)	(9)	35
10 (9)	20 (12)	20 (13)	(7)	50
(14)	(9)	10 (16)	30 (5)	40
45	20	30	30	

- $u_1 = 0$
- $u_1 + v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $u_2 + v_2 - 12 = 0$
- $u_2 + v_3 - 13 = 0$
- $u_3 + v_3 - 16 = 0$
- $u_3 + v_4 - 5 = 0$

Facile à résoudre ! !

On voit que si on pose $u_1 = 0$, toutes ces équations se réduisent à $u_i + v_j = c_{ij}$ pour les variables de base x_{ij}.

Illustration sur le problème d'électricité

35 (8)	(6)	(10)	(9)	35
10 (9)	20 (12)	20 (13)	(7)	50
(14)	(9)	10 (16)	30 (5)	40
45	20	30	30	

- $u_1 = 0$
- $v_1 = 8$
- $u_2 = 1$
- $v_2 = 11$
- $v_3 = 12$
- $u_3 = 4$
- $v_4 = 1$

On voit que si on pose $u_1 = 0$, toutes ces équations se réduisent à $u_i + v_j = c_{ij}$ pour les variables de base x_{ij}.

Calcul des coûts réduits

- Une fois qu'on a calculé les u_i et v_j le reste est facile ;
- En effet, les coûts réduits se calculent, pour toutes les variables hors base, par la formule suivante :

$$\overline{c}_{ij} = c_{ij} - u_i - v_j$$

- Dans l'exemple de la distribution d'électricité, on obtient :

$$\begin{array}{ll} \overline{c}_{12} = 6 - 11 + 6 = -5 & \overline{c}_{13} = 10 - 0 - 12 = -2 \\ \overline{c}_{14} = 9 + 0 - 1 = 8 & \overline{c}_{24} = 7 - 1 - 1 = 5 \\ \overline{c}_{31} = 14 - 4 - 8 = 2 & \overline{c}_{32} = 9 - 4 - 11 = -6 \end{array}$$

- Ici on cherche à minimiser, donc on choisit le coût réduit ayant la plus grande capacité à réduire le coût, soit $\overline{c}_{32}$. On fait donc entrer x_{32} dans la base.

Échange de variable

35 (8)	(6)	(10)	(9)	35
10 (9)	20 (12)	20 (13)	(7)	50
(14)	(9)	10 (16)	30 (5)	40
45	20	30	30	

- On fait entrer x_{32} dans la base ;
- Cela crée une boucle unique $x_{32} - x_{22} - x_{23} - x_{33}$;
- Les nœuds impairs de cette boucle sont x_{22} et x_{33}. La valeur de θ est la plus faible des deux, soit 10 ;
- On augmente les nœuds pairs (soient x_{32} et x_{23} de θ et on diminue d'autant les nœuds impairs ;
- Effectivement, on a échangé x_{33} avec x_{32}.

Échange de variable

35 (8)	(6)	(10)	(9)	35
10 (9)	10 (12)	30 (13)	(7)	50
(14)	10 (9)	(16)	30 (5)	40
45	20	30	30	

- On doit recalculer les coûts réduits
- On doit résoudre

$$\begin{array}{lll} u_1 = 0 & u_1 + v_1 = 8 & u_2 + v_1 = 9 \\ u_2 + v_2 = 12 & u_2 + v_3 = 13 & u_3 + v_2 = 9 \\ u_3 + v_4 = 5 & & \end{array}$$

- On doit ensuite calculer $\overline{c}_{ij} = c_{ij} - u_i - v_j$ pour toutes les variables hors-base.
- On trouve que les seules négatives sont

$$\overline{c}_{12} = -5, \overline{c}_{24} = -1, \overline{c}_{13} = -2,$$

- Donc x_{12} entre dans la base.

Échange de variable

35 (8)	(6)	(10)	(9)	35
10 (9)	10 (12)	30 (13)	(7)	50
(14)	10 (9)	(16)	30 (5)	40
45	20	30	30	

- On fait entrer x_{12} dans la base ;
- Cela crée une boucle unique $x_{12} - x_{22} - x_{21} - x_{11}$;
- Les nœuds impairs de cette boucle sont x_{22} et x_{11}. La valeur de θ est la plus faible des deux, soit 10 ;
- On augmente les nœuds pairs (soient x_{12} et x_{21} de θ et on diminue d'autant les nœuds impairs ;
- Effectivement, on a échangé x_{22} avec x_{12}.

Échange de variable

25 (8)	10 (6)	(10)	(9)	35
20 (9)	(12)	30 (13)	(7)	50
(14)	10 (9)	(16)	30 (5)	40
45	20	30	30	

- On doit recalculer de nouveau les coûts réduits
- On doit résoudre

$$
\begin{array}{lll}
u_1 = 0 & u_1 + v_1 = 8 & u_1 + v_2 = 6 \\
u_2 + v_1 = 9 & u_2 + v_3 = 13 & u_3 + v_2 = 9 \\
u_3 + v_4 = 5 & &
\end{array}
$$

- On doit ensuite calculer $\bar{c}_{ij} = c_{ij} - u_i - v_j$ pour toutes les variables hors-base.
- On trouve que le seul coût réduit négatif est
$$\bar{c}_{13} = -2$$
- Donc x_{13} entre dans la base.

Échange de variable

25 (8)	10 (6)	(10)	(9)	35
20 (9)	(12)	30 (13)	(7)	50
(14)	10 (9)	(16)	30 (5)	40
45	20	30	30	

- On fait entrer x_{13} dans la base ;
- Cela crée une boucle unique $x_{13} - x_{23} - x_{21} - x_{11}$;
- Les nœuds impairs de cette boucle sont x_{23} et x_{11}. La valeur de θ est la plus faible des deux, soit 25 ;
- On augmente les nœuds pairs (soient x_{13} et x_{21} de θ et on diminue d'autant les nœuds impairs ;
- Effectivement, on a échangé x_{11} avec x_{13}.

Échange de variable

(8)	10 (6)	25 (10)	(9)	35
45 (9)	(12)	5 (13)	(7)	50
(14)	10 (9)	(16)	30 (5)	40
45	20	30	30	

- On doit recalculer de nouveau les coûts réduits
- On doit résoudre

$$\begin{array}{lll} u_1 = 0 & u_1 + v_2 = 6 & u_1 + v_3 = 10 \\ u_2 + v_1 = 9 & u_2 + v_3 = 13 & u_3 + v_2 = 9 \\ u_3 + v_4 = 5 & & \end{array}$$

- On doit ensuite calculer $\overline{c}_{ij} = c_{ij} - u_i - v_j$ pour toutes les variables hors-base.
- On ne trouve aucun coût réduit négatif
- On a trouvé l'optimum !
- l'optimum est $z = 6*10 + 10*25 + 45*9 + 5*13 + 10*9 + 30*5 = 1020$.

Définition

- Un problème de transport pur achemine directement du producteur au consommateur ;
- Dans un problème de transbordement, on peut acheminer par des points intermédiaires du réseau ;
- On résout ce type de problème en les transformant en problèmes de transport purs.

Exemple de problème de transbordement

- Soit l'entreprise W, qui fabrique des jouets, l'une à Montpellier, l'autre à Douais. Celle de Montpellier peut en fabriquer 150 par jour, celle de Douais 200.
- Les jouets sont envoyé par la route aux magasins à Lyon et Brest. Les clients dans ces villes achètent 130 jouets.
- Du fait des coûts de transports moins élevés par rail, il peut être moins cher de faire passer les jouets par Nevers et/ou Castres. Les coûts d'acheminement sont les suivants :

	M	D	N	C	L	B
M	0	-	8	13	25	28
D	-	0	15	12	26	25
N	-	-	0	6	16	17
C	-	-	6	0	14	16
L	-	-	-	-	0	-
B	-	-	-	-	-	0

Transformation en problème de transport

Conclusion

- Les problèmes de transport, affectaction et transbordement sont des cas particuliers de LP, qu'on ne résout pas par le simplexe habituel.
- Il existe une méthode de résolution plus simple, non matricielle.
- Si les coût sont entiers, la solution est également entière, donc si on peut formuler un problème sous forme de transport, la solution en entier est également facilement calculable.

Conclusion générale sur le cours

- Ce cours est une introduction à la *recherche opérationnelle* ;
- C'est un domaine très important, dont le domaine d'application grandit chaque jour ;
- Récente pub d'IBM : 20% des containers arrivant aux USA sont vide !
- Récent résultat théoriques : par optimisation *convexe* on peut dans certains cas échantillonner plus efficacement qu'avec Shannon (Terence Tao, médaille Fields, UCLA 2008) *compressed sensing*.
- Peu de gens savent manier l'optimisation. J'espère que cette discipline vous sera utile.
- Tenez moi au courant !